



在上最佳化 Microsoft 工作負載的成本 AWS

# AWS 方案指引



# AWS 方案指引: 在上最佳化 Microsoft 工作負載的成本 AWS

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon 的商標和商業外觀不得用於任何非 Amazon 的產品或服務，也不能以任何可能造成客戶混淆、任何貶低或使 Amazon 名譽受損的方式使用 Amazon 的商標和商業外觀。所有其他非 Amazon 擁有的商標均為其各自擁有者的財產，這些擁有者可能附屬於 Amazon，或與 Amazon 有合作關係，亦或受到 Amazon 贊助。

# Table of Contents

|                                 |    |
|---------------------------------|----|
| 簡介 .....                        | 1  |
| 概觀 .....                        | 1  |
| 目標對象 .....                      | 1  |
| 本指南的使用方式 .....                  | 1  |
| 目標業務成果 .....                    | 3  |
| 成本最佳化旅程 .....                   | 4  |
| 最佳化成本的熱門建議 .....                | 6  |
| 概觀 .....                        | 6  |
| 熱門建議 .....                      | 6  |
| AWS 最佳化和授權評估 .....              | 8  |
| 概觀 .....                        | 8  |
| 評估選項 .....                      | 8  |
| 完整評估 .....                      | 9  |
| 範圍工作負載 .....                    | 9  |
| 收集資料 .....                      | 9  |
| 分析資料 .....                      | 10 |
| 規劃後續步驟 .....                    | 12 |
| 評估影響 .....                      | 13 |
| 後續步驟 .....                      | 14 |
| 其他資源 .....                      | 14 |
| Amazon EC2 上的 Windows .....     | 15 |
| 自動化停止和啟動排程 .....                | 15 |
| 概觀 .....                        | 15 |
| 案例研究 .....                      | 16 |
| 成本最佳化案例 .....                   | 17 |
| 成本最佳化建議 .....                   | 18 |
| 其他資源 .....                      | 29 |
| Windows 工作負載大小適中 .....          | 29 |
| 概觀 .....                        | 29 |
| 成本最佳化案例 .....                   | 30 |
| 成本最佳化建議 .....                   | 30 |
| 建議 .....                        | 38 |
| 其他資源 .....                      | 38 |
| 為 Windows 工作負載選取正確的執行個體類型 ..... | 39 |

|                                       |    |
|---------------------------------------|----|
| 概觀 .....                              | 39 |
| 成本最佳化建議 .....                         | 39 |
| 後續步驟 .....                            | 48 |
| 其他資源 .....                            | 49 |
| 為 Windows 和 SQL Server 工作負載提供授權 ..... | 49 |
| 概觀 .....                              | 49 |
| Amazon EC2 專用執行個體 .....               | 50 |
| AWS 授權選項 .....                        | 53 |
| 使用 Windows Server 授權 .....            | 53 |
| 成本最佳化案例 .....                         | 54 |
| 成本最佳化建議 .....                         | 60 |
| 其他資源 .....                            | 60 |
| 最佳化 Windows on Amazon EC2 的支出 .....   | 60 |
| 概觀 .....                              | 60 |
| 了解 Savings Plans .....                | 61 |
| 成本最佳化案例 .....                         | 66 |
| 成本最佳化建議 .....                         | 69 |
| 其他資源 .....                            | 71 |
| 使用 AWS 工具監控成本 .....                   | 71 |
| 概觀 .....                              | 71 |
| 成本最佳化建議 .....                         | 71 |
| 其他資源 .....                            | 74 |
| SQL Server .....                      | 75 |
| 選擇高可用性和災難復原解決方案 .....                 | 76 |
| 概觀 .....                              | 76 |
| SQL Server Always On 可用性群組 .....      | 77 |
| SQL Server Always On 容錯移轉叢集執行個體 ..... | 78 |
| SIOS DataKeeper .....                 | 80 |
| Always On 可用性群組 .....                 | 82 |
| 分散式可用性群組 .....                        | 83 |
| 日誌傳送 .....                            | 84 |
| AWS Database Migration Service .....  | 86 |
| AWS Elastic Disaster Recovery .....   | 86 |
| 成本比較 .....                            | 87 |
| 成本最佳化建議 .....                         | 90 |
| 其他資源 .....                            | 91 |

|                                       |     |
|---------------------------------------|-----|
| 了解 SQL Server 授權 .....                | 92  |
| 概觀 .....                              | 92  |
| AWS 授權選項 .....                        | 92  |
| 取得授權的成本影響 .....                       | 93  |
| 授權最佳化 .....                           | 93  |
| 成本最佳化建議 .....                         | 93  |
| 其他資源 .....                            | 38  |
| 為 SQL Server 工作負載選取正確的 EC2 執行個體 ..... | 98  |
| 概觀 .....                              | 99  |
| 成本比較 .....                            | 99  |
| 成本最佳化案例 .....                         | 100 |
| 成本最佳化建議 .....                         | 101 |
| 其他資源 .....                            | 105 |
| 合併執行個體 .....                          | 105 |
| 概觀 .....                              | 105 |
| 成本最佳化案例 .....                         | 105 |
| 成本最佳化建議 .....                         | 107 |
| 其他資源 .....                            | 107 |
| 比較 SQL Server 版本 .....                | 108 |
| 概觀 .....                              | 108 |
| 成本影響 .....                            | 108 |
| 成本最佳化建議 .....                         | 110 |
| 其他資源 .....                            | 114 |
| 評估 SQL Server 開發人員版本 .....            | 115 |
| 概觀 .....                              | 115 |
| 成本影響 .....                            | 115 |
| 其他資源 .....                            | 38  |
| 評估 Linux 上的 SQL Server .....          | 118 |
| 概觀 .....                              | 118 |
| 成本影響 .....                            | 119 |
| 成本最佳化建議 .....                         | 120 |
| 其他資源 .....                            | 121 |
| 最佳化 SQL Server 備份策略 .....             | 121 |
| 概觀 .....                              | 121 |
| 使用啟用 VSS 的快照進行伺服器層級備份 .....           | 122 |
| 使用的 SQL Server 備份 AWS Backup .....    | 124 |

|  |     |
|--|-----|
| 資料庫層級備份 .....                                | 126 |
| 成本最佳化建議 .....                                | 134 |
| 其他資源 .....                                   | 136 |
| 現代化 SQL Server 資料庫 .....                     | 136 |
| 概觀 .....                                     | 136 |
| 資料庫產品 .....                                  | 137 |
| Amazon RDS 和 Aurora 比較 .....                 | 137 |
| 成本最佳化建議 .....                                | 139 |
| 其他資源 .....                                   | 143 |
| 最佳化 SQL Server 的儲存 .....                     | 143 |
| 概觀 .....                                     | 143 |
| Amazon EBS 的 SSD 儲存類型、效能和成本 .....            | 144 |
| Amazon EBS 的一般 SSD 成本最佳化 .....               | 145 |
| 其他資源 .....                                   | 147 |
| 使用 Compute Optimizer 最佳化 SQL Server 授權 ..... | 147 |
| 概觀 .....                                     | 147 |
| 成本最佳化建議 .....                                | 148 |
| 設定運算最佳化工具 .....                              | 148 |
| 其他資源 .....                                   | 149 |
| 使用 Compute Optimizer 最佳化 SQL Server 大小 ..... | 150 |
| 概觀 .....                                     | 150 |
| 設定運算最佳化工具 .....                              | 150 |
| 其他資源 .....                                   | 151 |
| 檢閱 SQL Server 工作負載 Trusted Advisor 的建議 ..... | 151 |
| 概觀 .....                                     | 151 |
| 成本最佳化建議 .....                                | 151 |
| 設定 Trusted Advisor .....                     | 152 |
| 其他資源 .....                                   | 152 |
| 容器 .....                                     | 154 |
| 將 Windows 應用程式移至容器 .....                     | 154 |
| 概觀 .....                                     | 154 |
| 成本利益 .....                                   | 155 |
| 成本最佳化建議 .....                                | 156 |
| 後續步驟 .....                                   | 159 |
| 其他資源 .....                                   | 159 |
| 最佳化 Amazon ECS 上 AWS Fargate 任務的成本 .....     | 160 |

|   |     |
|---|-----|
| 概觀 .....                                    | 160 |
| 成本利益 .....                                  | 160 |
| 成本最佳化建議 .....                               | 160 |
| 後續步驟 .....                                  | 166 |
| 其他資源 .....                                  | 166 |
| 取得 Amazon EKS 成本的可見性 .....                  | 166 |
| 概觀 .....                                    | 166 |
| 成本利益 .....                                  | 166 |
| 成本最佳化建議 .....                               | 167 |
| 後續步驟 .....                                  | 170 |
| 其他資源 .....                                  | 170 |
| 使用 App2Container 複寫 Windows 應用程式 .....      | 170 |
| 概觀 .....                                    | 170 |
| 成本利益 .....                                  | 171 |
| 成本最佳化建議 .....                               | 172 |
| 後續步驟 .....                                  | 172 |
| 其他資源 .....                                  | 172 |
| 儲存 .....                                    | 173 |
| Amazon EBS .....                            | 173 |
| 將 Amazon EBS 磁碟區從 gp2 遷移至 gp3 .....         | 173 |
| 修改 Amazon EBS 快照 .....                      | 177 |
| 刪除未連接的 Amazon EBS 磁碟區 .....                 | 179 |
| Amazon FSx .....                            | 182 |
| 選擇正確的 SMB 檔案儲存體 .....                       | 182 |
| 在 Amazon FSx 中啟用重複資料刪除 .....                | 186 |
| 了解 FSx for Windows File Server 中的資料碎片 ..... | 188 |
| 了解 Amazon FSx 中的 HDD 磁碟區用量 .....            | 192 |
| 使用單一可用區域 .....                              | 194 |
| AWS Storage Gateway .....                   | 196 |
| Amazon S3 檔案閘道 .....                        | 196 |
| Amazon FSx 檔案閘道 .....                       | 196 |
| 成本影響 .....                                  | 196 |
| 成本最佳化建議 .....                               | 198 |
| 其他資源 .....                                  | 201 |
| Active Directory .....                      | 202 |
| Amazon EC2 上的自我管理 Active Directory .....    | 202 |

|                                     |     |
|-------------------------------------|-----|
| 概觀 .....                            | 202 |
| 成本影響 .....                          | 202 |
| 成本最佳化建議 .....                       | 203 |
| 其他資源 .....                          | 207 |
| AWS Managed Microsoft AD .....      | 207 |
| 概觀 .....                            | 207 |
| 成本影響 .....                          | 207 |
| 成本最佳化建議 .....                       | 207 |
| 其他資源 .....                          | 208 |
| AD Connector .....                  | 209 |
| 概觀 .....                            | 209 |
| 成本影響 .....                          | 209 |
| 成本最佳化建議 .....                       | 209 |
| 其他資源 .....                          | 210 |
| .NET .....                          | 211 |
| 重構現代 .NET 並移至 Linux .....           | 212 |
| 概觀 .....                            | 212 |
| 成本影響 .....                          | 212 |
| 成本最佳化建議 .....                       | 213 |
| 其他考量事項和資源 .....                     | 213 |
| 容器化 .NET 應用程式 .....                 | 214 |
| 概觀 .....                            | 214 |
| 成本影響 .....                          | 214 |
| 成本最佳化建議 .....                       | 215 |
| 其他資源 .....                          | 217 |
| 使用 Graviton 執行個體和容器 .....           | 218 |
| 概觀 .....                            | 218 |
| 成本影響 .....                          | 218 |
| 成本最佳化建議 .....                       | 220 |
| 其他資源 .....                          | 220 |
| 支援靜態 .NET Framework 應用程式的動態擴展 ..... | 221 |
| 概觀 .....                            | 221 |
| 成本影響 .....                          | 225 |
| 成本最佳化建議 .....                       | 226 |
| 其他資源 .....                          | 227 |
| 使用快取來減少資料庫需求 .....                  | 227 |

|                   |     |
|-------------------|-----|
| 概觀 .....          | 227 |
| 成本影響 .....        | 227 |
| 成本最佳化建議 .....     | 228 |
| 其他資源 .....        | 233 |
| 考慮無伺服器 .NET ..... | 234 |
| 概觀 .....          | 234 |
| 成本影響 .....        | 234 |
| 成本最佳化建議 .....     | 234 |
| 其他資源 .....        | 237 |
| 考慮專門建置的資料庫 .....  | 238 |
| 概觀 .....          | 238 |
| 成本影響 .....        | 240 |
| 成本最佳化建議 .....     | 242 |
| 其他資源 .....        | 244 |
| 後續步驟 .....        | 245 |
| 文件歷史紀錄 .....      | 246 |
| 詞彙表 .....         | 247 |
| # .....           | 247 |
| A .....           | 247 |
| B .....           | 250 |
| C .....           | 251 |
| D .....           | 254 |
| E .....           | 257 |
| F .....           | 259 |
| G .....           | 260 |
| H .....           | 261 |
| I .....           | 262 |
| L .....           | 264 |
| M .....           | 265 |
| O .....           | 269 |
| P .....           | 271 |
| Q .....           | 273 |
| R .....           | 273 |
| S .....           | 276 |
| T .....           | 279 |
| U .....           | 280 |

|         |           |
|---------|-----------|
| V ..... | 281       |
| W ..... | 281       |
| Z ..... | 282       |
| .....   | cclxxxiii |

# 在上最佳化 Microsoft 工作負載的成本 AWS

Bill Pfeiffer、Chase Lindeman 和 Kevin Sookhan , Amazon Web Services (AWS)

2024 年 10 月 ([文件歷史記錄](#))

## 概觀

本指南提供建議、最佳實務和策略，以協助最佳化 Microsoft 工作負載的成本 AWS。本指南也包含基礎 AWS 知識、成本最佳化技術和參考架構，協助您建置和自動化符合商業目標且符合成本效益且高效能的工作負載。統稱為 Microsoft on AWS Cost Optimization (MACO)。MACO 指引是由產業專家所開發，以真實案例為基礎。

本指南涵蓋下列 Microsoft 工作負載：

- Amazon Elastic Compute Cloud (Amazon EC2) 上的 Windows
- SQL Server
- 容器
- 儲存
- Active Directory
- .NET

## 目標對象

本指南適用於架構師、工程師、管理員、總監、CTOs、技術決策者和 AWS 合作夥伴。擁有的先前經驗，以及對 AWS 帳單、Microsoft 技術和 AWS 系統管理的基本了解，會很有幫助，但並非必要。

## 本指南的使用方式

您可以使用本指南來規劃和實作您的 MACO 雲端之旅。我們建議您從頭到尾閱讀本指南，以全面了解最佳化 Microsoft 工作負載成本的選項和方法 AWS。您可以根據組織的需求檢閱下列工作負載區段：

- [Amazon EC2 上的 Windows](#)
- [SQL Server](#)
- [容器](#)

- [儲存](#)
- [Active Directory](#)
- [.NET](#)

 Important

本指南提供的程式碼範例僅供示範使用。最佳實務是在生產環境中使用開發環境之前，先在開發環境中測試所有程式碼。實作任何程式碼之前，建議您以小批次方式測試程式碼，然後使用檢閱程式碼所產生的成本變更[AWS Cost Explorer](#)。這可協助您對邊緣案例和其他稍後可能遇到的問題進行故障診斷。

 Important

本指南中的定價範例是以發佈時的價格為基礎。價格可能變動。此外，您的成本可能會根據您的 AWS 區域、AWS 服務配額和與雲端環境相關的其他因素而有所不同。

# 目標業務成果

本指南可協助您和組織達成下列業務成果：

- 了解如何根據資源使用率、第三方授權和應用程式相依性，使用最佳化和授權評估 AWS (AWS OLA) 來評估和最佳化您目前的內部部署和雲端環境。
- 使用 AWS Microsoft Workloads 的現代化計算器來開發成本最佳化的商業案例。
- 最佳化特定 Microsoft 工作負載的成本，包括 Amazon Elastic Compute Cloud (Amazon EC2)、SQL Server、容器、儲存體、Active Directory 和 .NET 上的 Windows 工作負載。

# 成本最佳化旅程

雲端遷移旅程的範圍、時間和特定路徑取決於您的業務目標、技術需求和其他因素。本節提供雲端遷移旅程的範例，著重於使用的 [雲端財務管理 AWS](#)，並遵循 MACO 建議和最佳實務。您可以使用此範例來了解如何為 Microsoft 工作負載設計雲端遷移旅程。

下列高階任務說明組織可以採取的方法，以實作 MACO 建議和最佳實務：

- 建立標記策略並啟用使用者定義的成本分配標籤。如需詳細資訊，請參閱白皮書標記資源的 AWS 最佳實務。 [AWS](#)
- 根據應用程式、團隊或部門定義預算。如需詳細資訊，請參閱帳單和成本管理使用者指南中的 [使用管理您的 AWS Budgets](#) 成本。 AWS
- 執行最佳化和授權評估 AWS (AWS OLA) 以加速節省。如需詳細資訊，請參閱 AWS 文件中的 [AWS 最佳化和授權評估](#)。
- 使用為 Windows 和 SQL Server 工作負載帶來自己的授權 (BYOL) Amazon Elastic Compute Cloud Dedicated Hosts。如需詳細資訊，請參閱本指南的 [為 Windows 和 SQL Server 工作負載提供授權](#) 一節。
- 最佳化您的 SQL Server 授權 AWS。如需詳細資訊，請參閱本指南的 [了解 SQL Server 授權](#) 一節。
- 為 Windows 工作負載選取正確的執行個體類型。如需詳細資訊，請參閱本指南的 [為 Windows 工作負載選取正確的執行個體類型](#) 一節。
- 為 SQL 工作負載選取正確的執行個體類型。如需詳細資訊，請參閱本指南的 [為 SQL Server 工作負載選取正確的 EC2 執行個體](#) 一節。
- 將 Amazon Elastic Block Store (Amazon EBS) 從 gp2 遷移至 gp3。如需詳細資訊，請參閱本指南的 [將 Amazon EBS 磁碟區從 gp2 遷移至 gp3](#) 一節。
- 使用 EC2 Instance Scheduler 控制工作負載 AWS。如需詳細資訊，請參閱本指南的 [自動化停止和啟動排程](#) 一節。
- 使用 SQL Server Developer Edition 移除非生產工作負載的 SQL Server 成本。如需詳細資訊，請參閱本指南的 [評估 SQL Server 開發人員版本](#) 一節。
- 使用適用於 Amazon FSx for Windows File Server 的單一可用區域來開發和測試工作負載。如需詳細資訊，請參閱本指南的 [使用單一可用區域](#) 一節。
- 使用授權您的 Windows 工作負載 AWS Compute Optimizer。如需詳細資訊，請參閱本指南的 [Windows 工作負載大小適中](#) 一節。
- 使用 Savings Plans 最佳化 Windows on Amazon EC2 上的支出。如需詳細資訊，請參閱本指南的 [最佳化 Windows on Amazon EC2 的支出](#) 一節。

- 在 FSx for Windows File Server 上啟用重複資料刪除。如需詳細資訊，請參閱本指南的 [在 Amazon FSx 中啟用重複資料刪除](#) 一節。
- 對 FSx for Windows File Server 上的檔案系統使用資料分割。如需詳細資訊，請參閱本指南的 [了解 FSx for Windows File Server 中的資料碎片](#) 一節。
- 最佳化 SQL Server 備份策略。如需詳細資訊，請參閱本指南的 [最佳化 SQL Server 備份策略](#) 一節。
- 讓靜態 .NET 架構應用程式支援動態擴展。如需詳細資訊，請參閱本指南 [支援靜態 .NET Framework 應用程式的動態擴展的](#)。
- 使用無伺服器 .NET 微服務。如需詳細資訊，請參閱本指南的 [考慮無伺服器 .NET](#) 一節。
- 將您的 Windows 應用程式移至容器。如需詳細資訊，請參閱本指南的 [容器化 .NET 應用程式](#) 一節。
- 使用 [AWS Compute Optimizer](#) 將 Amazon Elastic Container Service (Amazon ECS) 上執行 AWS Fargate 的 Windows 容器授權。如需詳細資訊，請參閱本指南的 [啟用運算最佳化工具](#) 一節。
- 重構現代 .NET 並移至 Linux。如需詳細資訊，請參閱本指南的 [重構現代 .NET 並移至 Linux](#) 一節。
- 利用 Graviton 執行個體和容器。如需詳細資訊，請參閱本指南的 [使用 Graviton 執行個體和容器](#) 一節。
- 現代化 SQL Server 資料庫。如需詳細資訊，請參閱本指南的 [現代化 SQL Server 資料庫](#) 一節。
- 設計 Active Directory 基礎設施。如需詳細資訊，請參閱本指南的 [Active Directory](#) 一節。

如需使用 專注於雲端財務管理的客戶旅程的詳細資訊 AWS，請參閱 AWS 白皮書 [雲端財務管理功能](#)。

# 最佳化成本的熱門建議

## 概觀

成本最佳化是 [AWS Well-Architected Framework](#) 的支柱之一，它在您的雲端遷移計劃中扮演關鍵角色。您可以在本指南中找到成本最佳化的建議，但本節會指出影響程度最高的建議。您可以快速實作這些建議，這些建議將對您的組織產生重大影響。這些建議可協助您為整個成本最佳化工作奠定基礎。

## 熱門建議

下表列出最高影響成本最佳化的首要建議。「難以實作」資料欄會根據最容易實作的規模 (1) 對最難以實作的規模 (5) 來評分每個最佳化。「預估節省金額」欄顯示組織針對每個建議最佳化可以節省多少百分比的預估。

| 最佳化   | 難以實作 | 預估節省      |
|---|------|-----------|
| <a href="#">Windows 工作負載大小適中</a>                | 3    | 25%       |
| <a href="#">為 Windows 和 SQL Server 工作負載提供授權</a> | 3    | 30%       |
| <a href="#">評估 SQL Server 開發人員版本</a>            | 2    | 20%       |
| <a href="#">了解 SQL Server 授權</a>                | 2    | 高達 50%    |
| <a href="#">自動化停止和啟動排程</a>                      | 3    | 高達 40%    |
| <a href="#">為 Windows 工作負載選取正確的執行個體類型</a>       | 1    | 10–30%    |
| <a href="#">重構現代 .NET 並移至 Linux</a>             | 5    | 10–20%    |
| <a href="#">最佳化 Windows on Amazon EC2 的支出</a>   | 3    | 高達 20–40% |
| <a href="#">將 Amazon EBS 磁碟區從 gp2 遷移至 gp3</a>   | 4    | 高達 20%    |

**⚠ Important**

上表中的預估節省金額適用於每個技術網域，而不是帳戶內的整體 AWS 花費。例如，您可以在各種環境類型和大小中實作執行個體排程器，以改變潛在節省。預估值特別適用於 Amazon EC2 執行個體成本，並不表示其他執行個體的任何整體節省 AWS 服務。這些預估值是以計量提供，而不是保證。

MACO 專家可以更深入地討論成本最佳化。若要設定會議以深入了解您的使用案例，請聯絡您的客戶團隊或傳送電子郵件至 [optimize-microsoft@amazon.com](mailto:optimize-microsoft@amazon.com)。

# AWS 最佳化和授權評估

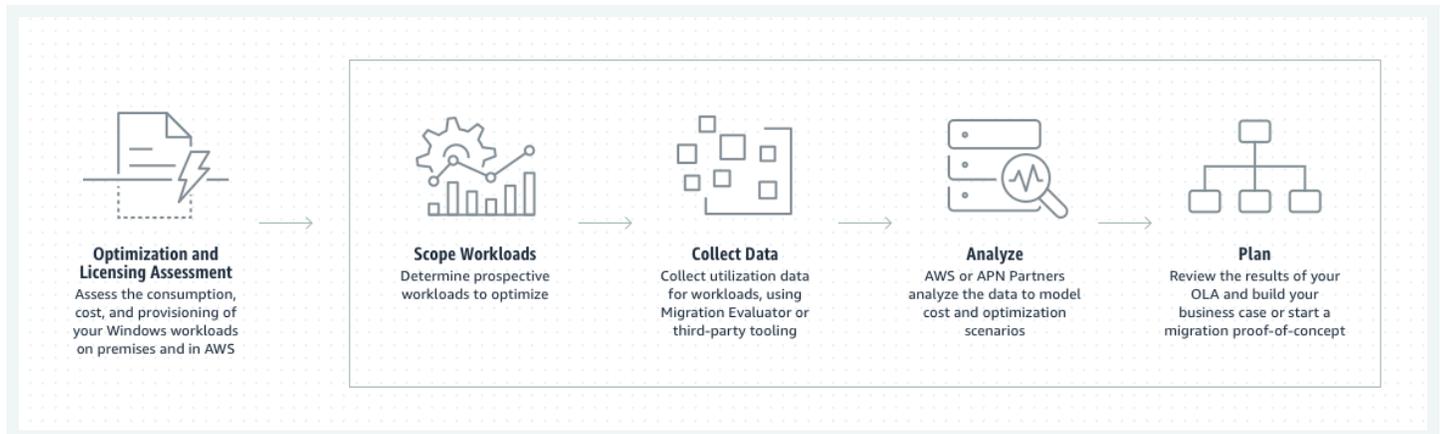
## 概觀

[AWS 最佳化和授權評估 \(AWS OLA\)](#) 可協助您根據資源使用率、第三方授權和應用程式相依性，評估和最佳化目前的內部部署和現有雲端環境。您可以使用 AWS OLA 來協助組織建置遷移和授權策略，以便在遷移到 AWS 或評估現有的 Microsoft 工作負載時，釋放成本節省 AWS。AWS OLA 也可以協助您達成下列目標：

- 了解現有的部署、應用程式效能和合約。
- 正確調整資源的大小。
- 開發通往的藍圖 AWS 雲端。
- 使用現有投資並僅支付您使用的項目，以降低或消除成本。

我們建議您將 AWS OLA 作為[成本最佳化旅程](#)的第一步。您可以使用 AWS Partner Network 來完成 AWS OLA。他們將協助您收集評估資料，並提供最佳化授權和執行個體成本的建議。

下圖提供評估程序的概觀。



## 評估選項

您可以在 AWS 上為 Microsoft 工作負載選擇兩個 OLA 選項 AWS：

- Lite 版本 – 在此使用案例中，您的所有工作負載都在 VMware 上。您可以 AWS 提供來自 [RVTools](#) 的輸出。然後，AWS 可以提供 1-5 天的周轉時間。此方法使用直接從 VMware vCenter 提取 point-in-time 資訊，來開發大小調整建議並提供隨需定價選項。

- 完整版本 – 在此使用案例中，您在不同的雲端提供者、實體伺服器 and 虛擬伺服器中執行混合環境。AWS 會使用作業系統代理程式來收集 14 到 30 天的用量資料。這可讓 根據您的應用程式使用模式 AWS，做出明智的執行個體規模調整決策。AWS 使用數個第三方工具，例如 Cloudamize 來完成分析。與其 AWS 合作 AWS Partner Network，以協助提供具有多種定價選項的最終總體擁有成本 (TCO) 評估，這些選項會考量定價模型和不同架構。

## 完整評估

完整的 AWS OLA 評估會在一小時的電話通話中啟動。在此呼叫期間，AWS 可協助您判斷支援遷移的最佳 AWS 基礎設施、選擇資料收集方法，以及建立完成的時間表。在組織中實作探索工具取決於資料收集方法、組織大小，以及組織用來管理其伺服器機群的工具。通常需要兩週的時間來收集用量資料。

完整的 AWS OLA 程序需要 30–45 天，且包含下列階段：

- 範圍工作負載
- 收集資料
- 分析資料
- 規劃後續步驟

## 範圍工作負載

首先，與您和您的團隊 AWS 合作，以確定評估的範圍。這通常依環境類型（例如，非生產和生產）細分。此範圍包含工作負載的位置。這可能是您要遷移到的工作負載 AWS、已在 上執行的工作負載 AWS（例如 Amazon EC2 的 AWS OLA），或是在其他雲端提供者中執行的工作負載。

## 收集資料

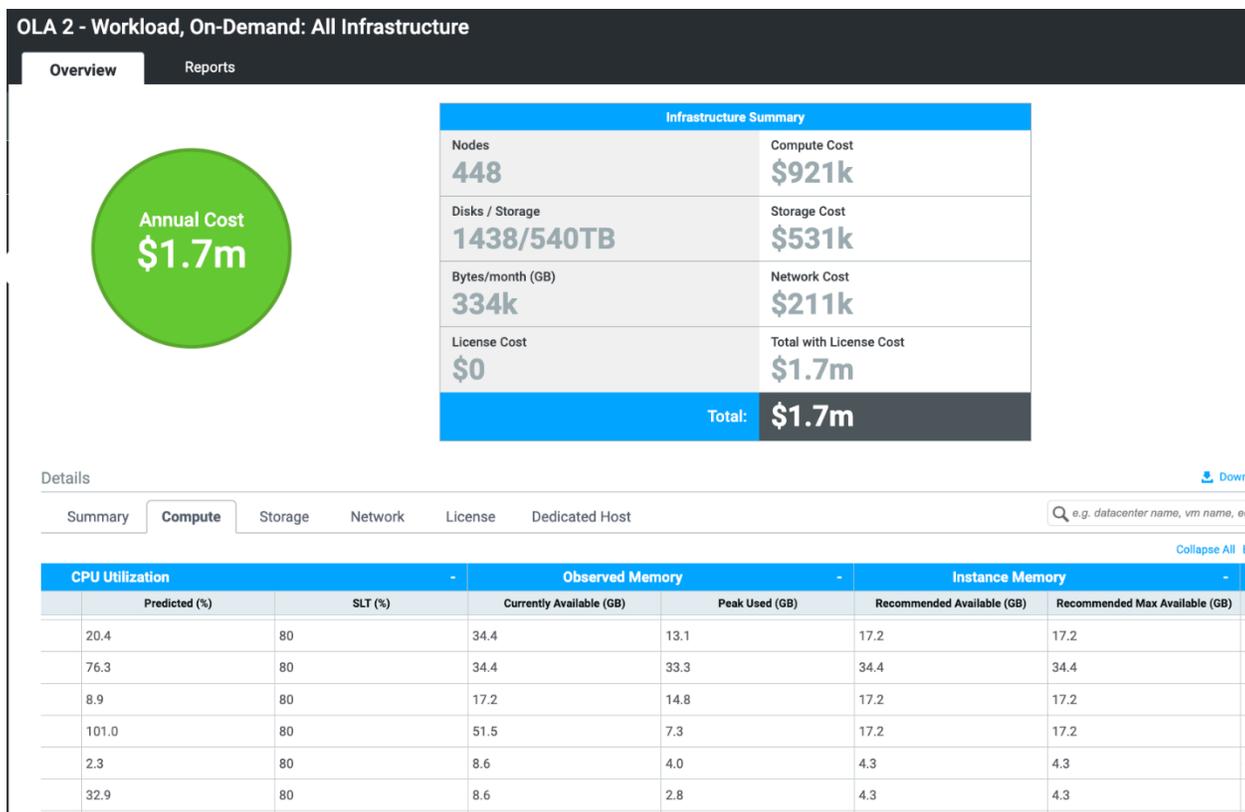
接下來，AWS 部署工具以協助資源探索，並從伺服器收集效能資料。此工具提供四種部署選項：

- 可查詢 Hypervisor 的工具（只需要 VMware vCenter 或 Hyper-V 登入資料）
- 可以部署在實體或虛擬機器上的代理程式
- 根據您的環境和作業系統，使用 SSH、Windows Remote Management (WinRM) 或 Windows Management Instrumentation (WMI) 進行無代理程式探索
- 平面檔案資料收集和分析

對於您的工具部署，您可以混合和比對每個選項，並合併結果。請務必確保您選擇的任何選項都不會讓您的 IT 資源承受壓力。AWS strives 會盡可能使評估程序成為統包。OLA 團隊和 Microsoft AWS 專業解決方案架構師除了簡短的通話來協助設定之外，還將準備總擁有成本 (TCO) 分析和建議以供審核。

分析 CPU 使用率、RAM 使用率、儲存輸送量、IOPS 和網路輸送量時，資料收集通常需要兩到三週的時間。理想情況下，此集合會在您業務月的尖峰時間（例如，end-of-month 財務報告期間）進行。AWS 想要擷取尖峰用量，因為這可為適當大小 AWS 的執行個體提供良好的統計樣本，同時仍然保證效能可以超過內部部署可用的效能。會 AWS 合併使用率指標與各種處理器世代的效能啟發，以準確鎖定指定工作負載所需的 CPU 和 RAM 數量。這些目標通常小於內部部署配置的目標。這不僅可以降低執行個體大小的運算成本，還可以最佳化授權成本。

下列儀表板檢視顯示可由評估擷取的基礎設施成本範例。

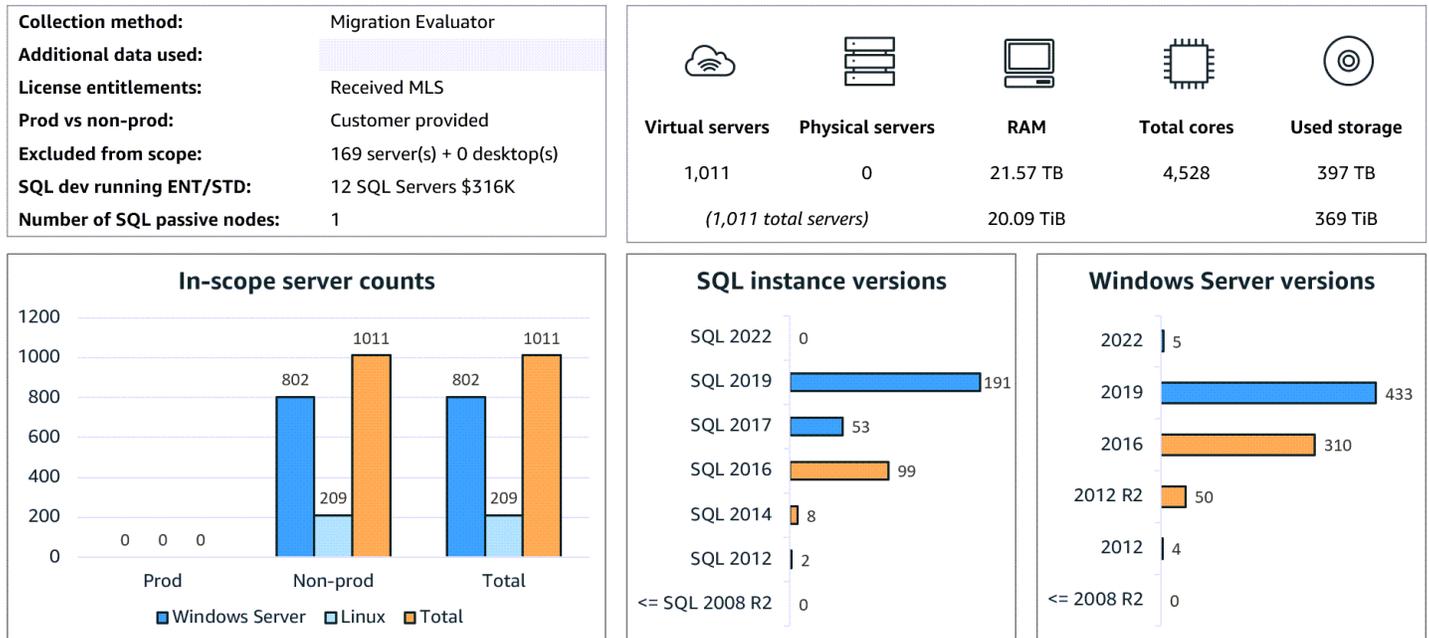


## 分析資料

AWS 會在資料收集完成後提供簡報報告。會 AWS 檢閱資料、摘要調查結果，然後針對內部部署用量和雲端遷移提出建議。您可以透過檢查整合機會、彈性增益（其中工作負載可以關閉或季節性調整）、正確 SKU 機會（例如，SQL Server Enterprise Edition 正在使用中，但資源需求和功能用量建議 SQL Server Standard Edition 適合）來降低運算和授權成本。對於由核心授權的 SQL Server 等產

品，將工作負載放置在更昂貴的運算執行個體中通常具有財務意義。也就是說，如果 CPU 設定檔和 RAM 與 vCPU 的比率提供淨效果，以減少包含授權和自攜授權 (BYOL) 使用案例的授權核心數量。

以下顯示以評估所收集的資料為基礎的範例分析。



常見的最佳化案例包括識別 AWS 資源最佳化機會和節省第三方授權。

AWS 資源最佳化機會的範例：

- 避免過度佈建尖峰用量。
- 避免過度指定和低用資源。
- 正確調整執行個體的大小，並遷移至最新一代的 EC2 執行個體。
- 移至受管資料庫，節省營運成本。

第三方授權節省的範例：

- 減少執行相同工作負載所需的核數。
- 排除不必要的 SQL Server Enterprise 版本和附加元件套件。
- 移除 zombie 伺服器並取代過時的硬體。
- 使用 BYOL 和包含授權的選項來減少未來的商業協議。
- 將現代化為開放原始碼和雲端原生解決方案。

## 規劃後續步驟

最後，AWS 會使用所收集的效能資料來估計特定工作負載大小和成本。AWS 也可以在您的範圍環境中進行彙總，並提供量化分析。這可協助您判斷最佳選項是內部部署重新整理或遷移至 AWS。您可以使用 OLA 結尾提供的 TCO 分析摘要（如下列範例所示）AWS 來建置雲端經濟商業案例。

|  | <b>Option 1:<br/>Amazon EC2 shared</b> | <b>Option 1a:<br/>Amazon EC2 shared +<br/>power management</b> | <b>Option 2:<br/>Amazon EC2 mixed</b> | <b>Option 2a:<br/>Amazon EC2 mixed +<br/>power management</b> |
|--|--|--|---------------------------------------|---|
| <i>Option details: compute</i>                                 | 100% Reserved Instances (RIs)          | RIs + on-demand power management                               | 100% RIs                              | RIs + on-demand power management                              |
| <i>Option details: Microsoft licenses</i>                      | WS LI and SQL BYOL                     | WS LI and SQL BYOL   | WS BYOL or LI+SQL BYOL                | WS BYOL or LI+SQL BYOL  |
| <b>Compute costs<sup>1</sup></b>                               |  |  |                                       |   |
| Year 1 compute cost  | \$414,546                              | \$482,623  | \$504,019                             | \$513,941   |
| Year 1 vendor license included cost                            | \$392,858                              | \$244,415  | \$9,804                               | \$4,783   |
|  | <b>\$807,404</b>                       | <b>\$727,038</b>   | <b>\$513,823</b>                      | <b>\$518,724</b>  |
| <i>Total compute savings in year 1, compared to Option 1</i>   | —                                      | 10% (\$80,366)   | 36% (\$293,581)                       | 36% (\$288,680)   |
| <b>Storage and networking costs<sup>2</sup></b>                |  |  |                                       |   |
| Annual estimated storage cost                                  | \$336,494                              | \$336,494  | \$336,494                             | \$336,494   |
| Annual estimated networking cost                               | \$41,455                               | \$41,455   | \$41,455                              | \$41,455  |
|  | <b>\$377,949</b>                       | <b>\$377,949</b>   | <b>\$377,949</b>                      | <b>\$377,949</b>  |
| <b>Microsoft license costs**</b>                               |  |  |                                       |   |
| WS/CIS annual Software Assurance (SA) + current SPLA/Subs cost | \$0                                    | \$0  | \$0                                   | \$0   |
| WS/CIS license + SA + SPLA/Subs true-up cost                   | \$0                                    | \$0  | \$0                                   | \$0   |
| SQL annual SA + current SPLA/Subs cost                         | \$0                                    | \$0  | \$0                                   | \$0   |
| SQL license SA + current SPLA/Subs true-up cost                | \$0                                    | \$0  | \$0                                   | \$0   |
|  | <b>\$0</b>                             | <b>\$0</b>   | <b>\$0</b>                            | <b>\$0</b>  |
| <b>Total estimated costs</b>                                   | <b>\$1,185,353</b>                     | <b>\$1,104,987</b>   | <b>\$891,772</b>                      | <b>\$896,673</b>  |
| <i>Annual TCO savings in year 1, compared to Option 1</i>      | —                                      | 7% (\$80,366)  | 25% (\$293,581)                       | 24% (\$288,680)   |

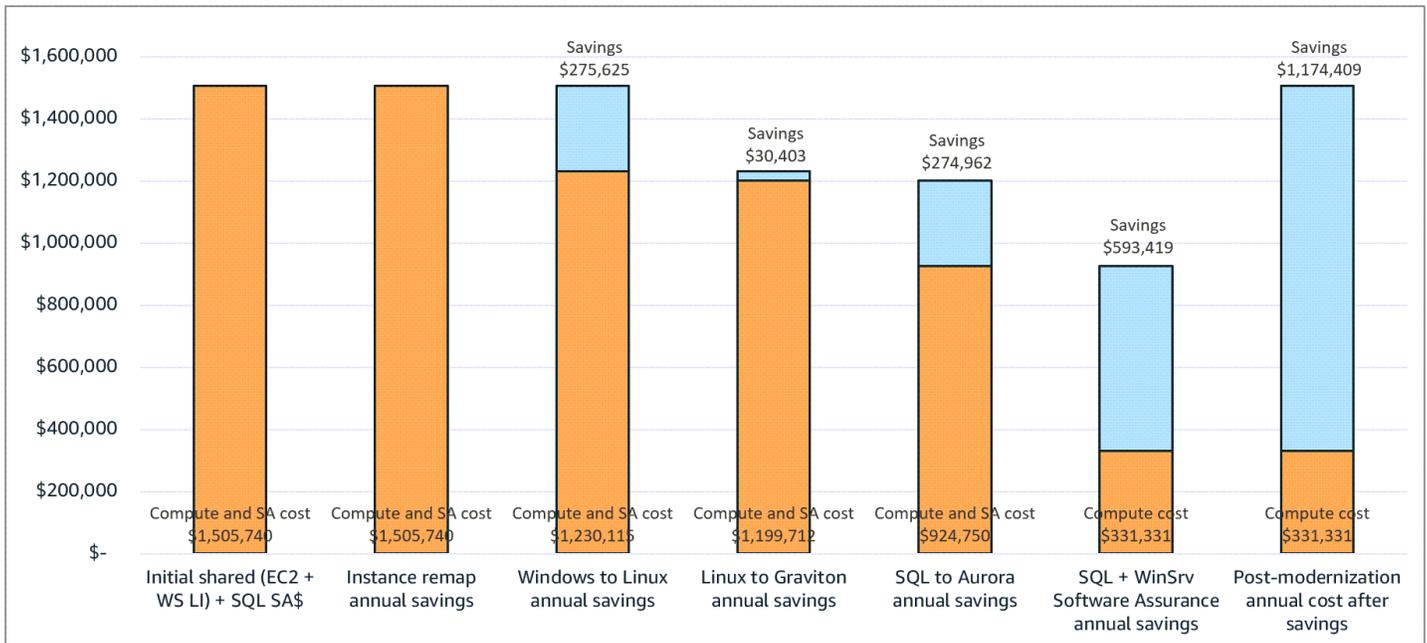
<sup>1</sup> Pricing model used: 3-year, no upfront RI

<sup>2</sup> Software Assurance and true-up costs provided by Microsoft

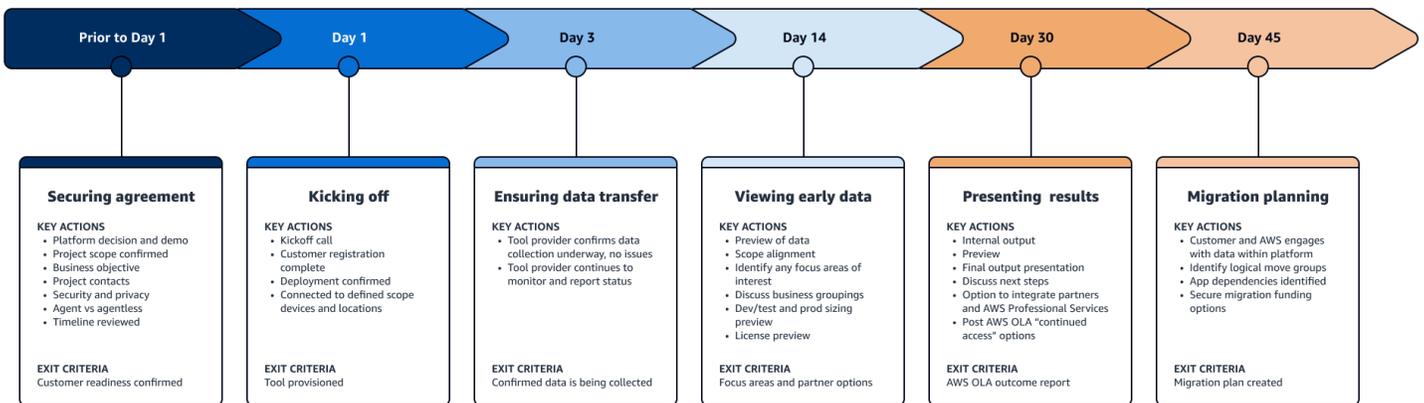
OLA AWS 也透過提出下列建議，來深入了解現代化可能對您現有工作負載造成的影響：

- 移至 Linux 作業系統。
- 新增 ARM 處理器的應用程式支援 (AWS Graviton)。
- 將 SQL Server 工作負載移至 Amazon Aurora。
- 將 Windows 和 SQL Server 工作負載移至開放原始碼技術，以移除軟體保證。

下圖顯示可透過現代化技術達成的成本節省，例如從 Windows 移至 Linux 或從 SQL Server 移至 Aurora。



整個 AWS OLA 程序從開始到結束大約需要 45 天。下圖顯示範例時間軸。



如果您擁有純 VMware 環境，並且可以提供來自 RVTtools 的輸出，則可以將此時間表縮短為一週。此外，AWS 可以分析包含資產和使用率資料的平面檔案，例如 CPU 平均值、CPU 峰值、RAM 平均值和 RAM 峰值。

## 評估影響

平均客戶通常會從合適的規模調整工作中降低成本 20–30%。根據用量資料，正確的大小會將來源工作負載與大小最佳的 AWS 執行個體相符。這些調整大小適中不僅可降低 AWS 環境的每月成本，而且經常在組織中的其他位置節省成本。例如，取得 20–30% 的 Windows 或 SQL Server 授權可以減少 Microsoft 的下一個校正，或釋放其他 line-of-business 應用程式的授權。SQL Server 工作負載的合併和正確調整大小通常是實現最顯著財務收益之處。

AWS 可協助您將系統分類為現代化儲存貯體。某些系統是舊版的，在財務上無法接觸，而其他系統可能會現代化為容器或無伺服器應用程式，而這些應用程式可實現最大的節省。與 AWS 團隊的對話會從雲端啟用的廣義主題，轉移到更具體的討論，說明特定工作負載應如何及為何進行現代化。AWS 也可協助您探索潛在的創新機會。

## 後續步驟

如果您要開始在內部部署環境或 上執行的 Microsoft 工作負載的成本最佳化旅程 AWS，請與您的 AWS 客戶團隊聯絡，並請求 AWS OLA。AWS 團隊成員可以回答您的問題，並協助您決定 AWS OLA 最終是否適合您和您的組織。或者，您也可以[線上請求 AWS OLA](#)。

## 其他資源

- [AWS 最佳化和授權評估 \(AWS 文件\)](#)
- [AWS re : Invent 2022 - 如何在 \(ENT205\) \(YouTube\) 上 AWS 節省成本並最佳化 Microsoft 工作負載](#)  
YouTube

# Amazon EC2 上的 Windows

[Amazon Elastic Compute Cloud \(Amazon EC2\)](#) 是高度靈活且可擴展的雲端運算平台，非常適合執行 Windows 工作負載。您可以使用 Amazon EC2，在安全、可靠、高可用性和可調整的基礎設施上部署、管理和擴展 Windows Server 工作負載 AWS 雲端。請考慮在 Amazon EC2 上執行 Windows 工作負載的下列主要優點：

- 可擴展性 – Amazon EC2 可讓您輕鬆擴展 Windows 工作負載，以因應不斷變化的需求。您可以快速建立新的 EC2 執行個體來處理增加的需求，並在不再需要執行個體時輕鬆終止執行個體。您只需為實際使用的資源付費。
- 彈性 – Amazon EC2 上的 Windows 支援各種執行個體類型，這些執行個體類型旨在滿足從一般用途執行個體到記憶體或運算最佳化執行個體的各種工作負載需求。此彈性可確保您可以為特定 Windows 型應用程式選擇最佳的執行個體類型，將效能最大化並將成本降至最低。
- 安全性 – 為您的 Windows 工作負載 AWS 提供多層安全性，包括網路防火牆、資料加密和安全存取控制。這表示您可以信任您的應用程式和資料受到保護，同時仍可完全控制您的安全設定和組態。
- 成本效益 – pay-as-you-go 定價模式可讓您僅支付使用的資源，無需在硬體和軟體中預先投資。此模型也可讓您最佳化成本、減少資本支出，並提高營運效率。對於各種規模的企業而言，這是理想的定價模式。

本指南的本節涵蓋下列主題：

- [自動化停止和啟動排程](#)
- [Windows 工作負載大小適中](#)
- [為 Windows 工作負載選取正確的執行個體類型](#)
- [為 Windows 和 SQL Server 工作負載提供授權](#)
- [最佳化 Windows on Amazon EC2 的支出](#)
- [使用 AWS 工具監控成本](#)

## 自動化停止和啟動排程

### 概觀

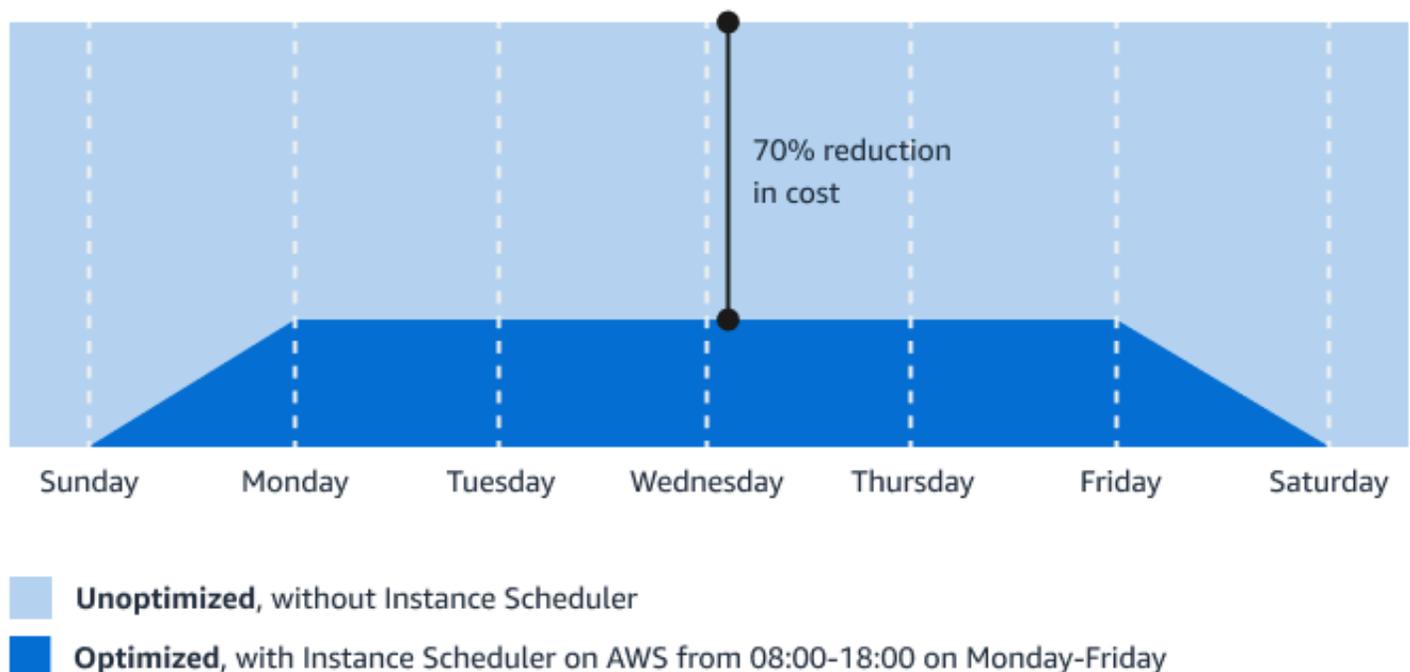
[上的執行個體排程器 AWS](#) 可以透過自動啟動和停止 [Amazon EC2](#) 和 [Amazon Relational Database Service \(Amazon RDS\)](#) 執行個體，協助您降低營運成本。如果您讓所有執行個體持續以完全使用率執

行，最終可能會支付未使用的資源。開啟的執行個體排程器 AWS 可讓您在不需要執行個體的時段關閉執行個體，例如非上班時間、週末或使用量低的其他時段。這可能會導致隨著時間的推移大幅節省成本。

上的執行個體排程器 AWS 也提供跨帳戶執行個體排程、自動標記，以及使用命令列界面或 [AWS Systems Manager](#) 維護時段來設定排程或期間的功能。這些功能可協助您更有效且準確地在不同專案或團隊中追蹤和分配成本。

## 案例研究

考慮在 AWS 生產環境中使用執行個體排程器的公司範例，每天在營業時間外自動停止執行個體。如果此公司以完全使用率保留其所有執行的執行個體，則這些執行個體最多可以節省 70% 的成本，而這些執行個體只在正常上班時間才需要。下表顯示每週使用率如何從 168 小時減少為 50 小時。

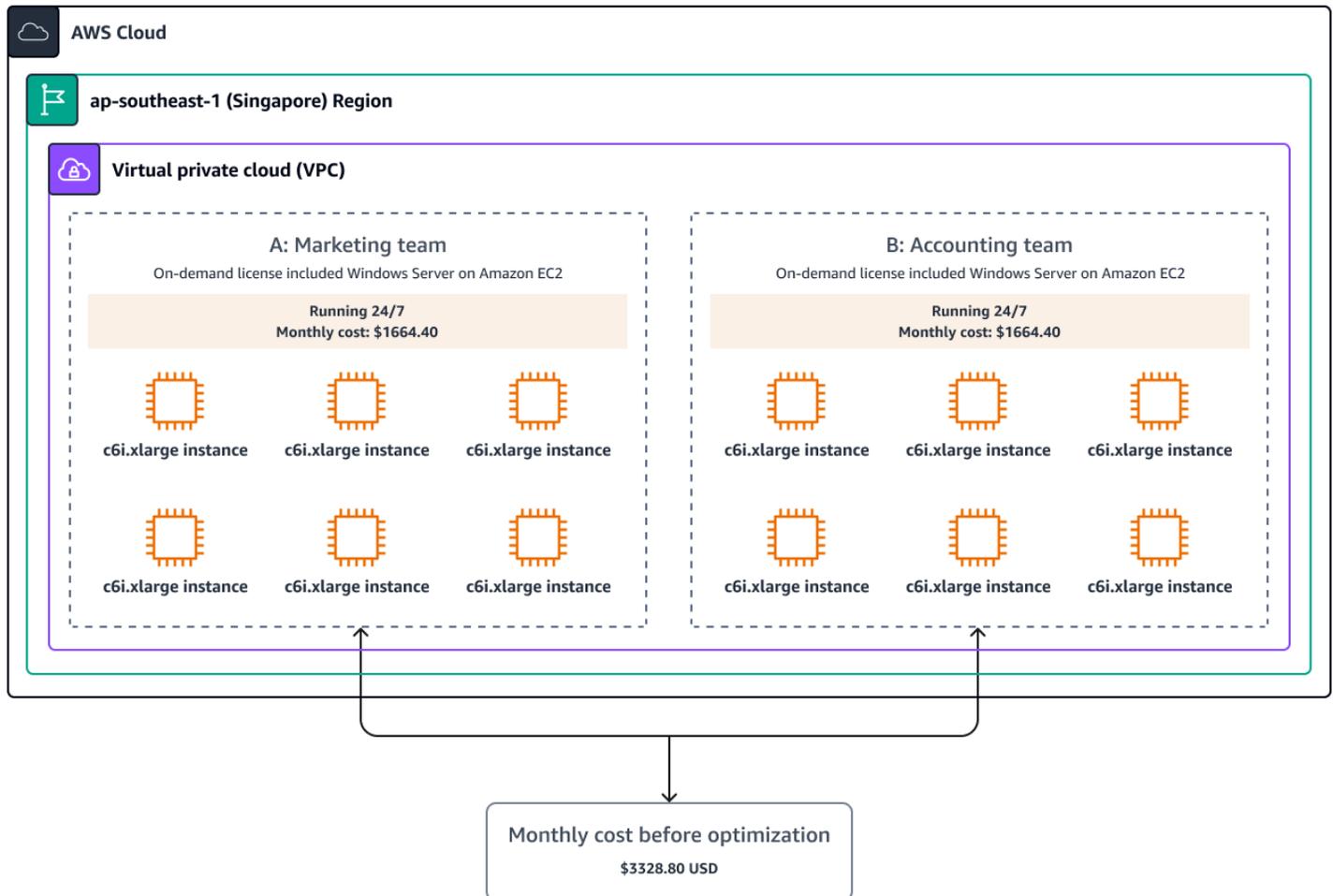


請考慮另一個範例。電力公用事業公司 Jamaica Public Service Company Limited (JPS) 將其資料庫遷移至 Amazon RDS。現在，JPS 使用 Amazon EC2 託管 API 服務並執行其他應用程式。對於 JPS，上的執行個體排程器 AWS 成為管理非生產環境的關鍵工具。JPS 使用上的執行個體排程器 AWS 來降低開發成本，並根據團隊需求和工作排程管理 EC2 執行個體。這有助於 JPS 將成本降低 40%。如需詳細資訊，請參閱 AWS 案例研究 [Jamaica Public Service 有效率地遷移至雲端，使用 AWS 執行個體排程器將成本降低 40%](#)。

## 成本最佳化案例

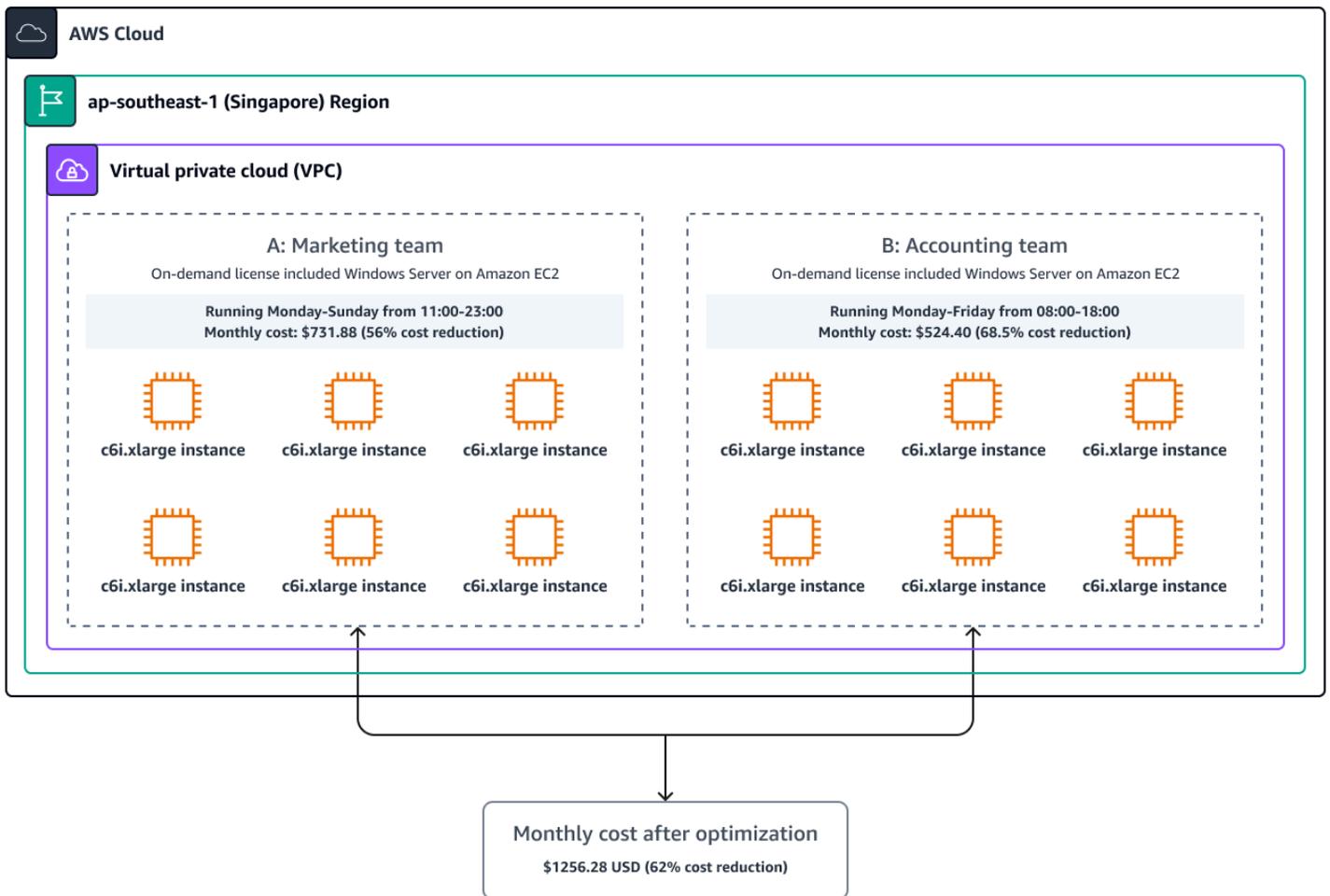
下列範例案例有助於說明使用執行個體排程器的成本優勢 AWS。在此案例中，新加坡一家主要零售公司在 Amazon EC2 上部署兩個 Windows 環境。行銷團隊會使用稱為工作負載 A 的第一個環境，在商店開放時分析即時的商店內交易。第二個環境稱為工作負載 B，保留給會計團隊，只在正常上班時間內運作。考慮到目前的使用模式，這兩個環境（全年無休）目前的操作排程並不理想，並且需要最佳化以降低公司的營運成本。

下圖顯示最佳化之前的每月成本。



例如，3 月有 31 天，其中 23 天是工作日。如果行銷團隊在上使用執行個體排程器 AWS，並僅在需要時才操作其執行個體（即每個月 321 小時，而不是每月 730 小時），他們可能會每月節省 932.52 美元。這相當於營運成本降低 56%。會計團隊也會遇到顯著的優勢，其執行個體使用時間從每月 730 小時降至 230 小時。這會導致 1,140 USD 或 68.5% 的折扣。公司每月可節省總計 2,072.52 USD（相當於 62% 的折扣），或每年 24,870.24 USD。

下圖顯示最佳化後的每月成本。



### Note

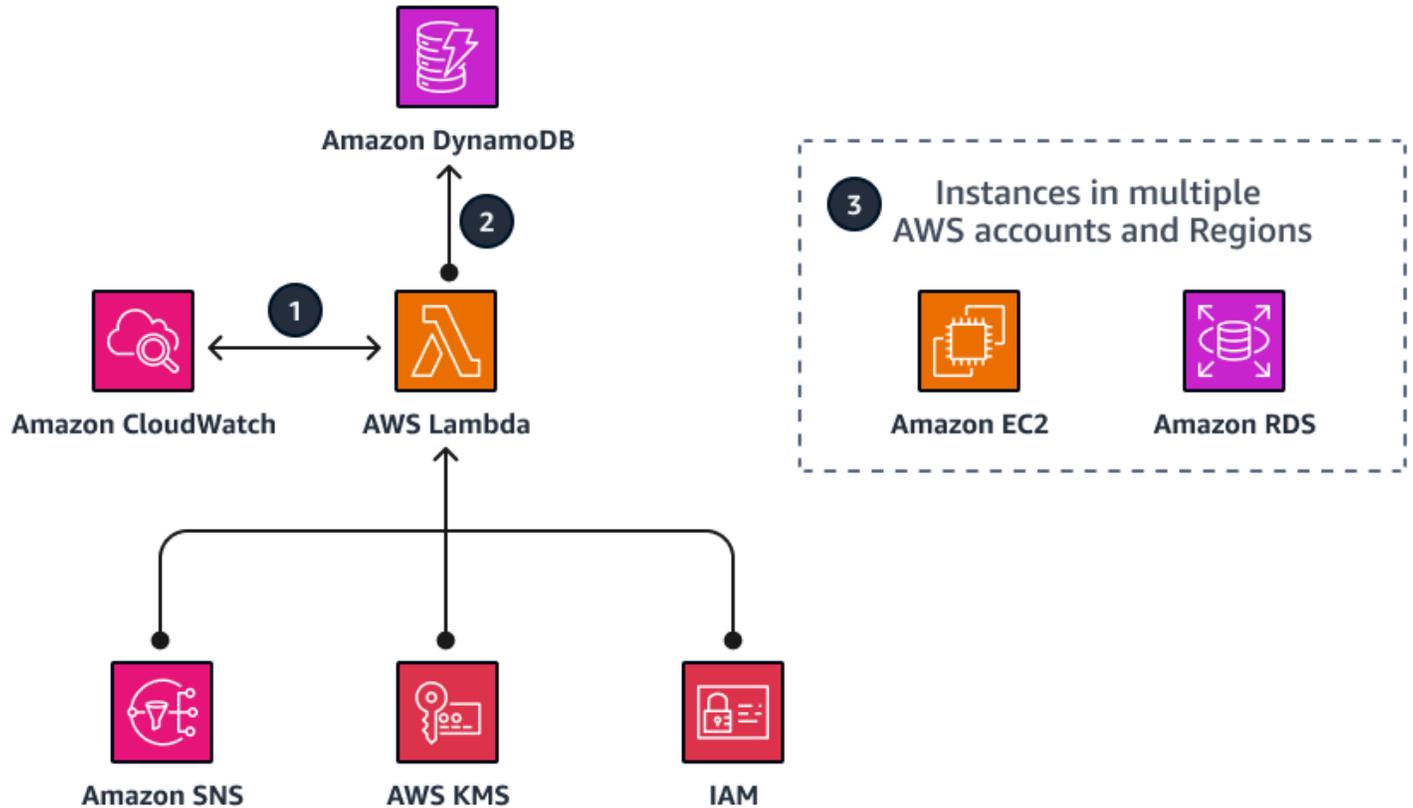
此範例的定價是在 2023 年 3 月 [AWS 定價計算工具](#) 使用決定。

## 成本最佳化建議

本節說明如何根據 AWS 上一個成本最佳化案例區段中涵蓋的範例案例來部署和設定執行個體排程器。建議您在上使用執行個體排程器，採取下列後續步驟來最佳化成本 AWS：

1. 啟動執行個體排程器堆疊
2. 設定期間
3. 設定排程
4. 標記執行個體

下列架構圖顯示執行個體排程器堆疊 AWS 雲端 在 中建立的內容。



圖表顯示下列工作流程步驟：

1. AWS CloudFormation 範本會以您定義的間隔設定 Amazon CloudWatch 事件。此事件會叫用 AWS Lambda 函數。在組態期間，您可以定義 AWS 區域 和 帳戶。您也可以定義上的執行個體排程器 AWS 用來將排程與適用的 Amazon EC2 執行個體、Amazon RDS 執行個體和叢集建立關聯的自訂標籤。
2. 排程組態值存放在 Amazon DynamoDB 中，Lambda 函數會在每次執行時擷取這些值。然後，您可以將自訂標籤套用至適用的執行個體。
3. 在執行個體排程器的初始組態期間，您可以定義標籤索引鍵來識別適用的 Amazon EC2 和 Amazon RDS 執行個體。建立排程時，您指定的名稱會用作標籤值，以識別您要套用至已標記資源的排程。

## 啟動執行個體排程器堆疊

本節說明如何啟動執行個體排程器的 CloudFormation 堆疊 AWS。

**Note**

您需負責執行執行個體排程器 AWS 服務時使用的成本 AWS。自 2023 年 1 月起，在 us-east-1 區域中使用預設設定執行此解決方案的成本約為每月 9.90 USD 的 Lambda 費用，或者如果您有 Lambda 免費方案每月用量額度，則費用會較低。如需詳細資訊，請參閱《AWS 解決方案程式庫》中[執行個體排程器在 AWS 實作指南](#)上的成本一節。

若要啟動執行個體排程器堆疊，請完成下列步驟。

1. 登入 [AWS Management Console](#) 並選擇 [啟動解決方案](#)（可下載範本）以啟動 instance-scheduler-on-aws.template CloudFormation 範本。

**Note**

您也可以將[下載範本](#)作為自有實作的起點。

2. 根據預設，範本會在美國東部（維吉尼亞北部）區域啟動。若要在不同區域中啟動執行個體排程器，請使用主控台導覽列中的區域選擇器。

**Note**

此範例使用亞太區域（新加坡）區域。

3. 在建立堆疊頁面的先決條件 - 準備範本區段中，確認已選取範本就緒選項。在範本來源區段中，確認已選取 Amazon S3 URL 選項。
4. 確認正確的範本 URL 位於 Amazon S3 URL 文字方塊中，然後選擇下一步。
5. 在指定堆疊詳細資訊頁面上，為您的解決方案堆疊指派名稱。如需有關命名字元限制的資訊，請參閱 AWS Identity and Access Management ([IAM](#)) 文件中的 [IAM 和 STS 限制](#)。本指南中範例的堆疊名稱稱為 MyInstanceScheduler。

**Note**

堆疊名稱不能包含超過 28 個字元。

6. 在參數下，檢閱範本的參數並視需要修改。
7. 選擇 Next (下一步)。在 Configure stack options (設定堆疊選項) 頁面，選擇 Next (下一步)。

8. 在檢閱頁面上，檢閱並確認設定。選取確認範本將建立 IAM 資源的方塊。
9. 選擇建立以部署堆疊。

## 設定期間

部署 CloudFormation 範本之後，解決方案會建立 DynamoDB 資料表，其中包含範例期間規則和排程，您可以用此資料表做為建立自訂期間規則和排程的參考。如需期間組態範例，請參閱 [文件上執行個體排程器](#) AWS 的範例排程。

若要完成此案例的步驟，您必須產生對應於每個工作負載的期間，並滿足其特定需求。例如：

```
Period 1 (Workload A):
  Name: retail-hours
  Days: Monday to Sunday
  Hours: 1100 - 2300
Period 2 (Workload B):
  Name: office-hours
  Days: Monday to Friday
  Hours: 0800 - 1800
```

若要設定期間，請完成下列步驟：

1. 登入 [DynamoDB 主控台](#)，並確保您位於您在 上啟動執行個體排程器的 CloudFormation 範本的相同區域 AWS。
2. 在導覽窗格中，選擇資料表，然後選取名為 ConfigTable 的資料表。
3. 選擇探索資料表項目。
4. 若要建立上班時間的期間，請選取上班時間項目的期間。
5. 在編輯項目頁面上，將開始時間的值變更為 0800，結束時間變更為 1800。保留工作日的預設值。

### Note

開始時間和結束時間值決定何時應啟動和停止執行個體，而工作日值則決定此排程適用於一週中的哪些天（此範例為週一至週五）。

6. 選擇 Save changes (儲存變更)。
7. 若要複製辦公時間期間並用它來建立新的零售時間期間，請選取辦公時間項目的期間。然後，從動作功能表中，選擇複製項目。
8. 修改屬性以符合您的需求。下列屬性用於滿足範例案例的需求：

```
type: period
name: retail-hours
begintime: 11:00
description: Retail hours
endtime: 23:00
weekdays: mon-sun
```

9. 選擇建立項目。

10. 在 DynamoDB ConfigTable 中，識別您剛在項目清單中列出的兩個期間。

## 設定排程

在執行個體排程器開啟的內容中 AWS，排程是指一或多個期間和相關時區的應用程式。然後，這些排程會以標籤的形式指派給您的執行個體。本節說明如何建立兩個排程（如下所示），以適應兩個範例工作負載的不同時間模式，然後將排程與您在上一節中建立的期間建立關聯。

```
Schedule 1:
  Name: singapore-office-hours
  Period: office-hours
  Timezone: Asia/Singapore
Schedule 2:
  Name: singapore-retail-hours
  Period: retail-hours
  Timezone: Asia/Singapore
```

若要建立和設定排程，請完成下列步驟：

1. 登入 [DynamoDB 主控台](#)，並確保您位於您在 上啟動執行個體排程器的 CloudFormation 範本的相同區域 AWS。
2. 在導覽窗格中，選擇資料表，然後選取名為 ConfigTable 的資料表。
3. 選擇探索資料表項目。
4. 若要複製英國辦公時間排程，並用它來為您的辦公時間建立新的排程（新加坡辦公時間，在此範例中），請選取 uk-office-hours 項目的排程。然後，從動作功能表中，選擇複製項目。
5. 修改屬性以符合您的需求。下列屬性用於滿足範例案例的需求：

```
type: schedule
name: singapore-office-hours
description: Office hours in Singapore
```

```
periods: office-hours  
timezone: Asia/Singapore
```

6. 選擇建立項目。
7. 重複步驟 4-6，使用下列屬性值建立新加坡零售時數的排程：

```
type: schedule  
name: singapore-retail-hours  
description: Retail hours in Singapore  
periods: retail-hours  
timezone: Asia/Singapore
```

8. 在 DynamoDB ConfigTable 中，識別您建立的兩個排程和兩個期間。

## 標記 執行個體

建立排程後，您必須使用標籤將排程配置給您想要使用的特定執行個體。您可以使用 [中的標籤編輯器](#) [AWS Resource Groups](#) 來產生標籤，並將標籤指派給您的 Amazon EC2 執行個體。

1. 登入 [AWS Management Console](#)，並確保您位於先前啟動 CloudFormation 範本的相同區域。
2. 開啟 [資源群組主控台](#)。在導覽窗格中，展開標記，然後選擇標籤編輯器。
3. 在尋找要標記的資源區段中，針對區域選擇您的區域。針對資源類型，選擇 Amazon EC2 或 Amazon RDS。此案例著重於工作負載 A 中的 Amazon EC2 執行個體。行銷團隊正在使用新加坡區域中的工作負載 A。此工作負載的資源已標記部門金鑰和行銷值。您可以使用此標籤來搜尋執行個體。
4. 選擇搜尋資源。
5. 從搜尋結果清單中選取要包含在排程中的執行個體，然後選擇管理所選資源的標籤。
6. 在編輯所有所選資源的標籤區段中，選擇新增標籤，將執行個體排程器排程標籤新增至您的 EC2 執行個體。您可以使用符合排程的標籤索引鍵和值（先前在 DynamoDB 中建立）。
7. 對於標籤金鑰，新增排程。在標籤值 中輸入 singapore-retail-hours。
8. 選擇檢閱和套用標籤變更。
9. 若要將標籤套用至您選取的所有 EC2 執行個體，請選擇套用變更至所有選取的執行個體。
10. 針對您想要套用的任何其他排程重複步驟 3-9。

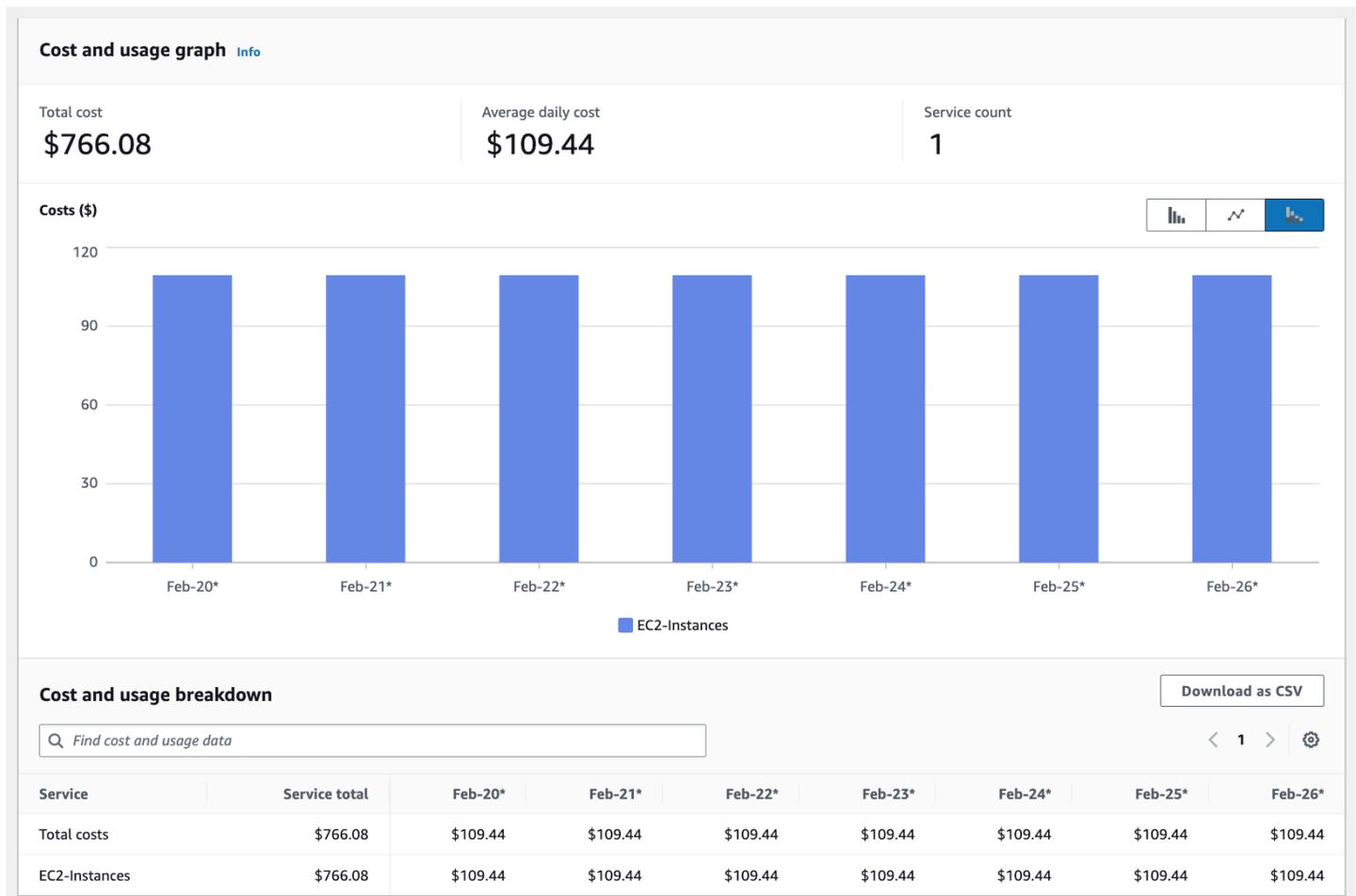
## 驗證結果

建議您使用 [AWS Cost Explorer](#) 來衡量使用執行個體排程器的成本優勢 AWS。您可以使用 Cost Explorer 執行下列動作：

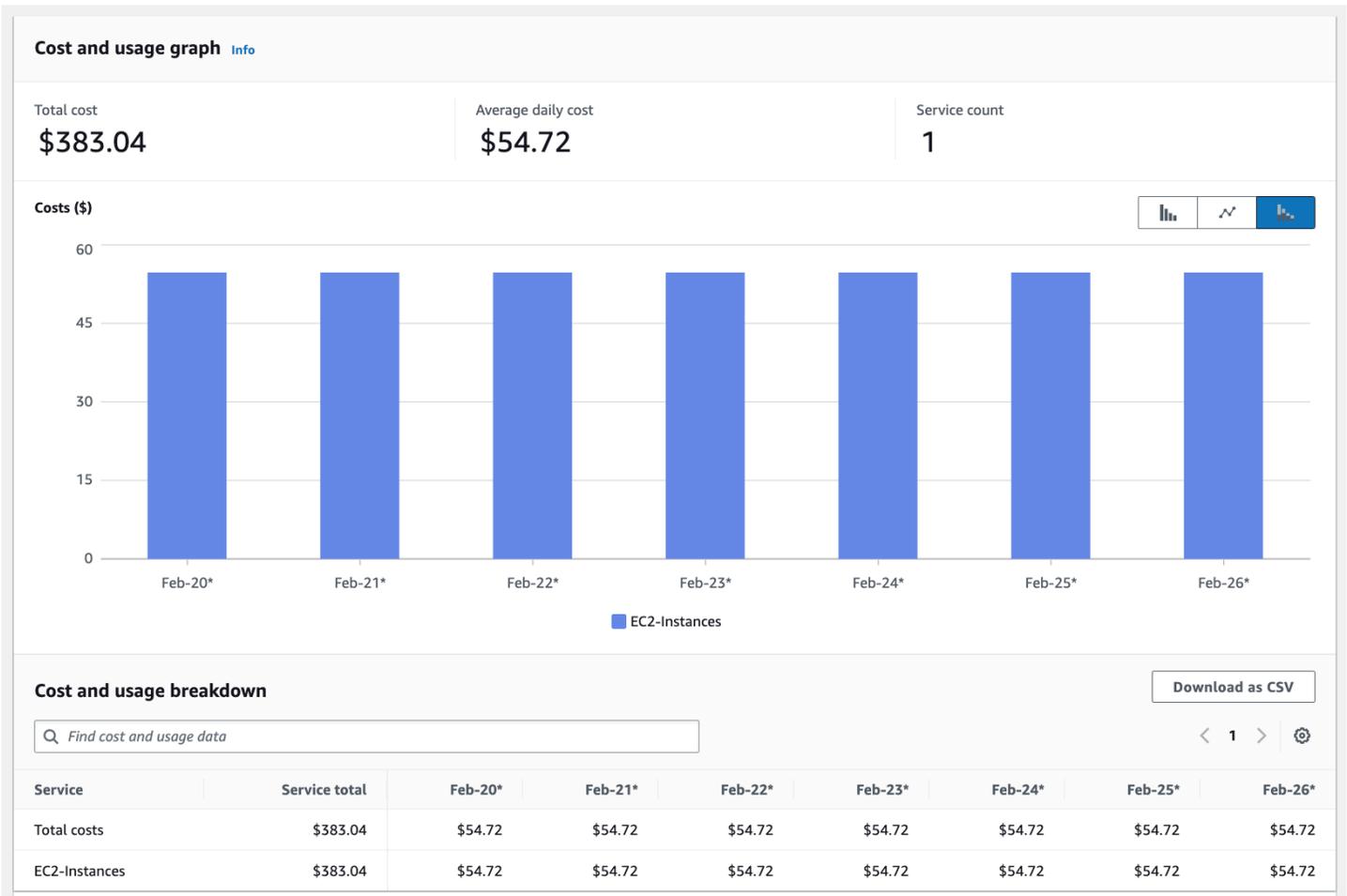
- 檢視和分析與 EC2 執行個體相關的成本，包括執行個體排程器管理的執行個體。
- 依標籤篩選 Cost Explorer 檢視，讓您可以專注於特定工作負載，並使用執行個體排程器取得節省成本的精細檢視。
- 深入了解使用執行個體排程器對財務的影響。
- 識別進一步成本最佳化的機會，並做出資料驅動型決策以最佳化您的 AWS 支出。

下表說明在最佳化前七天（週一至週日）內使用執行個體排程器操作工作負載 A 和工作負載 B 的成本。

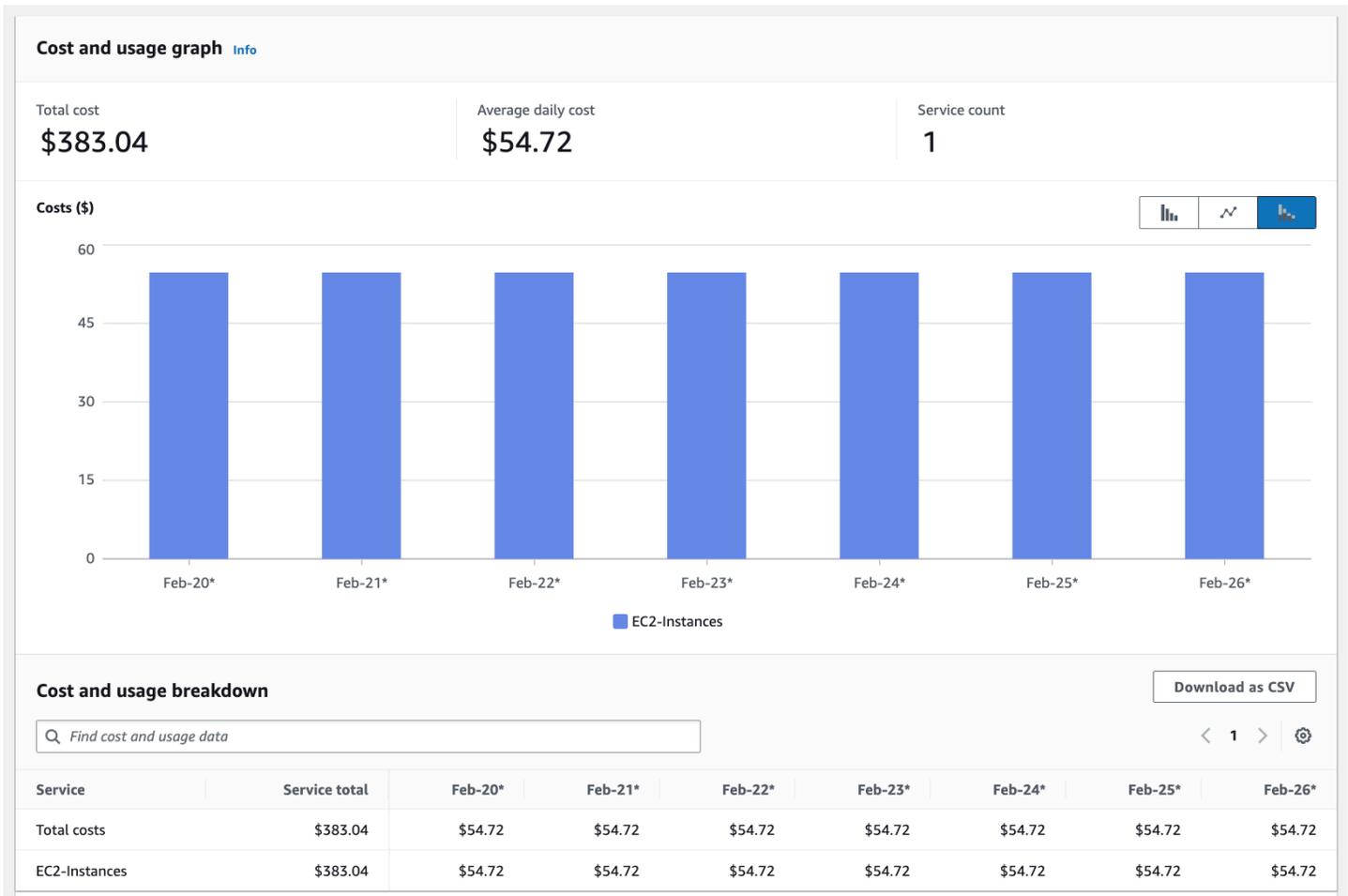
### 工作負載 A 和 B 的合併總費用



### 工作負載 A 費用

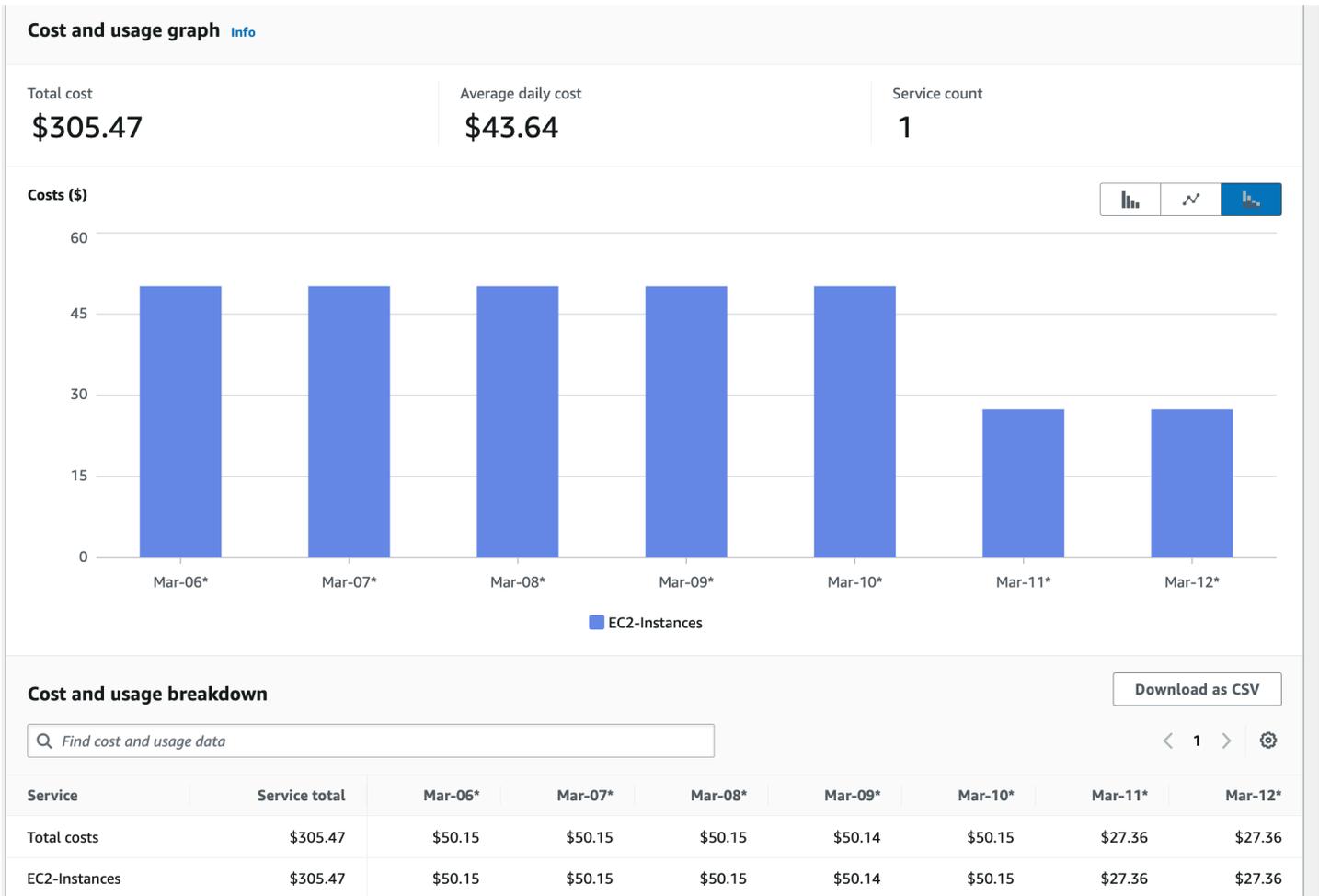


## 工作負載 B 費用

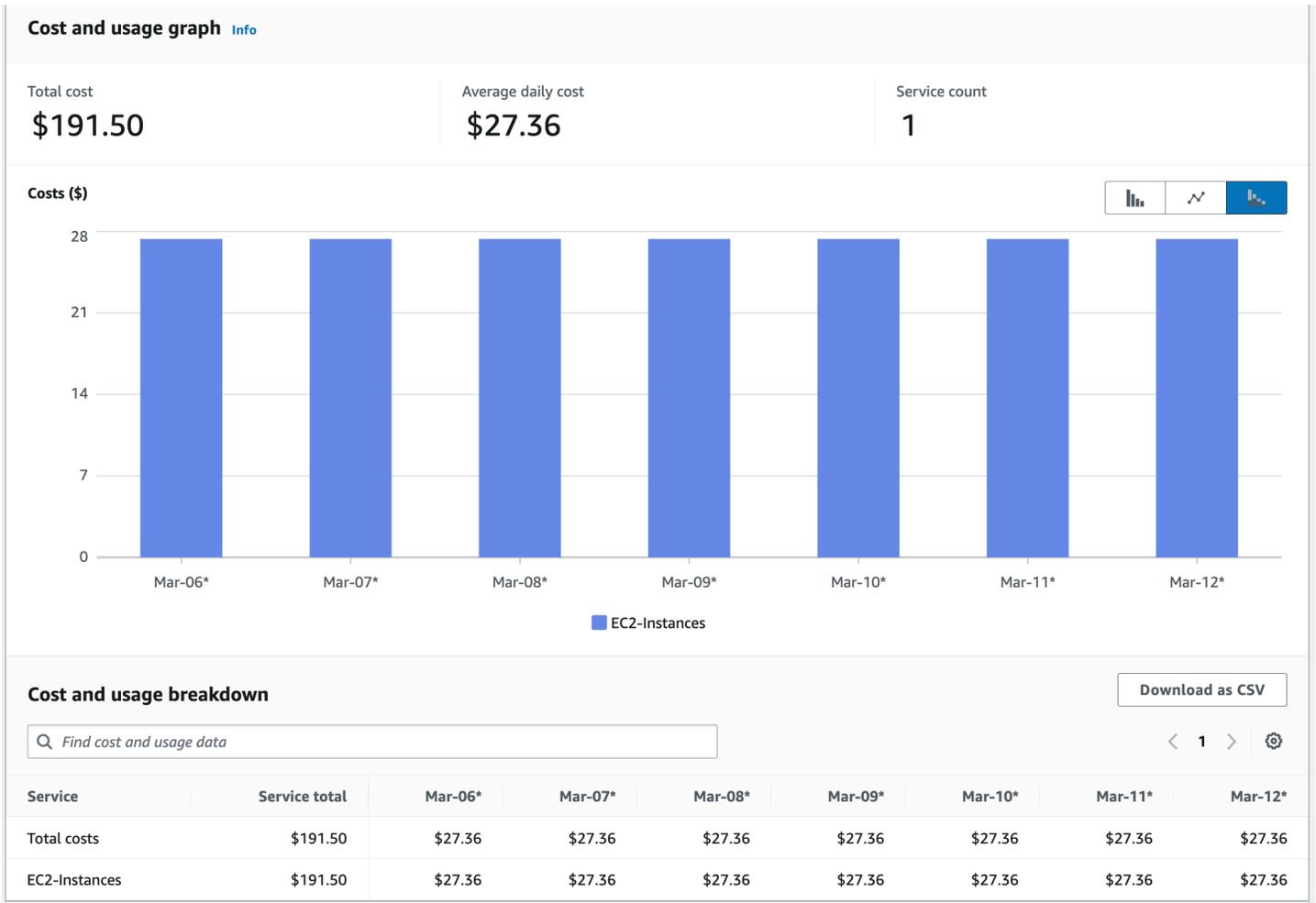


在此案例中，Cost Explorer 會顯示實作執行個體排程器所產生的成本降低 AWS。下表說明最佳化後七天（週一至週日）內工作負載 A 和工作負載 B 的操作成本。

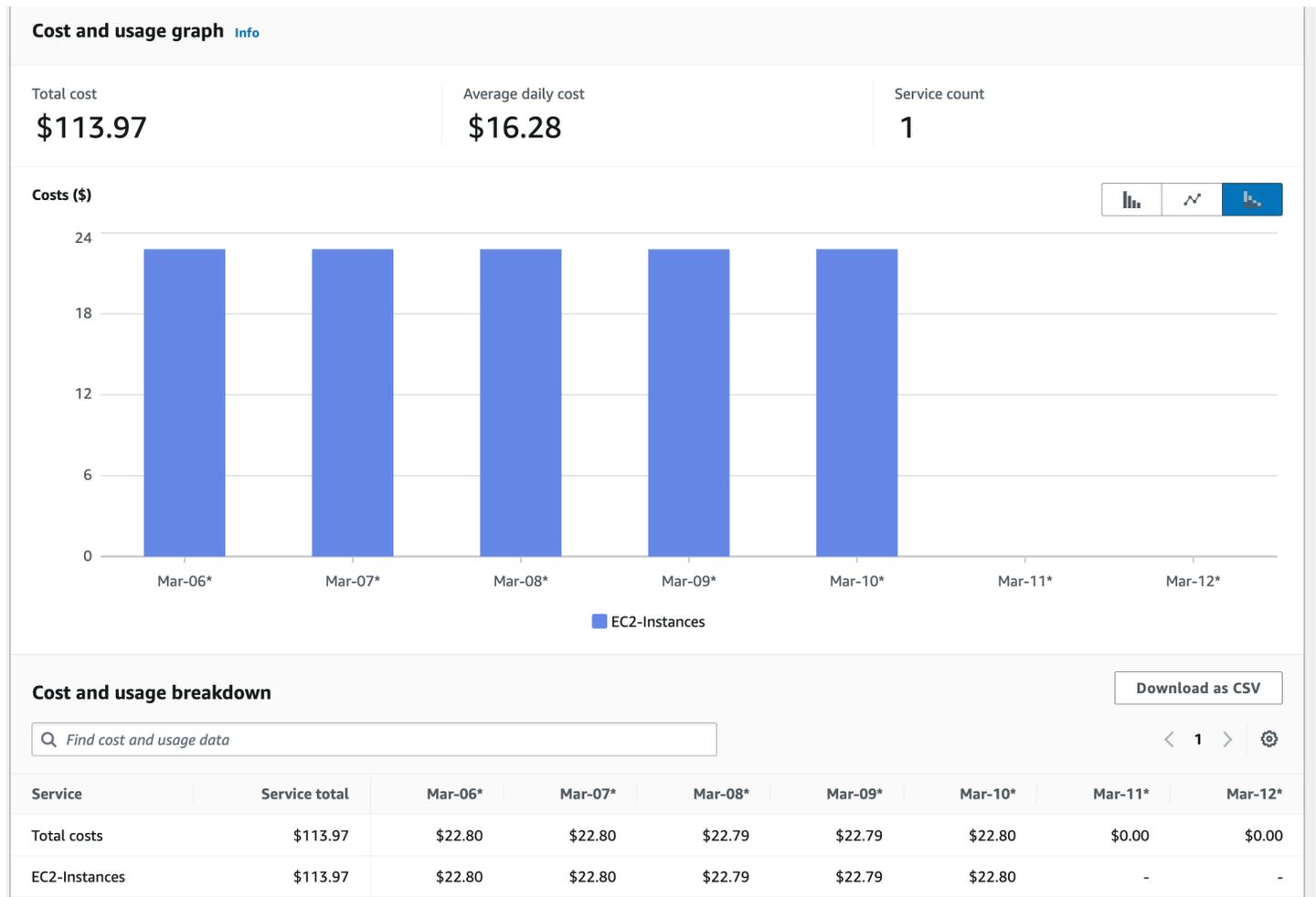
工作負載 A 和 B 的合併總費用



## 工作負載 A 費用



## 工作負載 B 費用



## 其他資源

- [自動化啟動和停止 AWS 執行個體](#) ( AWS 文件上的執行個體排程器 )
- [返回基本概念：使用執行個體排程器控制 Amazon EC2 和 Amazon RDS 資源成本](#) (YouTube)
- [標記您的 AWS 資源](#) ( 標記 AWS 資源使用者指南 )
- [使用 分析成本 AWS Cost Explorer](#)(AWS 帳單與成本管理 文件 )

## Windows 工作負載大小適中

### 概觀

適當調整大小是最有效的節省成本工具之一。AWS 提供各種方法來收集適當調整大小的資訊，包括使用 [AWS 最佳化和授權評估 \(AWS OLA\)](#) 來檢閱潛在的工作負載，以及使用 [AWS Cost Explorer](#) 來檢閱現有的工作負載。

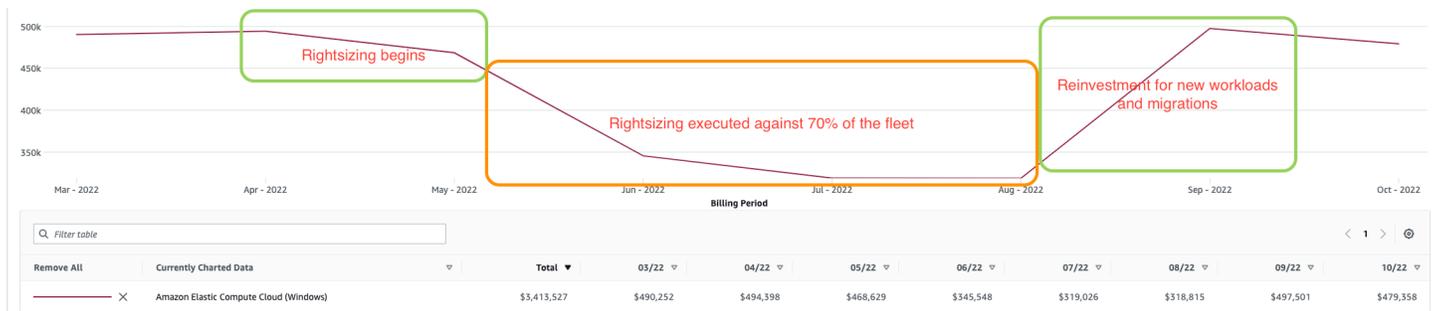
本節說明如何使用 [AWS Compute Optimizer](#) 來識別 Amazon EC2 適當調整大小的機會。Compute Optimizer 有助於防止下列資源類型的過度佈建和佈建不足 AWS：

- [Amazon Elastic Compute Cloud \(Amazon EC2\)](#) 執行個體類型
- [Amazon Elastic Block Store \(Amazon EBS\)](#) 磁碟區
- 上的 [Amazon Elastic Container Service \(Amazon ECS\)](#) 服務 AWS Fargate
- [AWS Lambda](#) 根據 [Amazon CloudWatch](#) 提供的使用率資料函數

## 成本最佳化案例

測量適當調整大小的有效性可能具有挑戰性，因為適當調整大小的工作可以導致特定應用程式、團隊或整個組織。例如，假設一個組織將數千個執行個體遷移到其中 AWS，其中 90% 的機群包含 Windows 工作負載。組織可以使用 Compute Optimizer 來分析其機群，並探索其帳戶和間大量的過度佈建 AWS 區域。然後，他們可以使用 [AWS Systems Manager 自動化](#)，透過多個維護時段正確調整機群的大小。因此，組織會針對 70% 的機群，設法調整適當大小的執行個體類型，並節省 35% 的成本。

以下儀表板說明此範例組織策略性地實作 Compute Optimizer 的正確大小建議，因此在幾個月內實現的節省。其目標是盡可能有效率地操作其現有工作負載，以便從即將合約結束的主機代管資料中心繼續停滯遷移。



## 成本最佳化建議

建議您採取下列後續步驟，使用 Compute Optimizer 來最佳化成本：

- 啟用運算最佳化工具
- 啟用 Windows 節點的記憶體指標集合
- 使用 Compute Optimizer 建議
- 標記執行個體以正確調整大小
- 啟用成本分配標籤以使用 AWS 帳單工具

- 使用 AWS Systems Manager 自動化實作正確的大小調整建議
- 考慮替代調整大小方法
- 在 Cost Explorer 中檢閱成本前後

## 啟用運算最佳化工具

您可以在組織或單一帳戶層級啟用 [Compute Optimizer](#) AWS Organizations。整個組織的組態會針對所有成員帳戶的整個機群中新的和現有的執行個體提供持續報告。這可讓適當調整大小成為週期性活動，而不是 point-in-time 活動。

### 組織層級

對於大多數組織而言，使用 Compute Optimizer 最有效率的方式是在組織層級。這可提供您組織的多帳戶和多區域可見性，並將資料集中到一個來源以供檢閱。若要在組織層級啟用此功能，請執行下列動作：

1. 使用具有 **必要許可** 的角色登入您的 [Organizations 管理帳戶](#)，並選擇加入此組織中的所有帳戶。您的組織必須 [啟用所有功能](#)。
2. 啟用管理帳戶後，您可以登入帳戶、查看所有其他成員帳戶，以及瀏覽他們的建議。

#### Note

最佳實務是設定 Compute Optimizer 的 [委派管理員帳戶](#)。這可讓您執行最低權限原則。如此一來，您就可以將對組織管理帳戶的存取降到最低，同時仍提供整個組織的服務存取。

### 單一帳戶層級

如果您以成本高的帳戶為目標，但無法存取 AWS Organizations，您仍然可以為該帳戶和區域啟用 Compute Optimizer。若要了解選擇加入程序，請參閱 Compute Optimizer 文件中的 [入門 AWS Compute Optimizer](#)。

## 啟用 Windows 節點的記憶體指標集合

記憶體指標為 Compute Optimizer 提供在組織中提供正確調整大小建議所需的基本指標。這是因為在提供建議之前，正在分析 CPU、記憶體、網路和儲存體。

若要將記憶體指標從 Windows EC2 執行個體傳遞至 Compute Optimizer，您必須啟用 CloudWatch 代理程式，並設定每 60 秒收集一次記憶體指標。搭配 CloudWatch 使用記憶體指標無需額外費用。

## 啟用 CloudWatch 代理程式並設定記憶體指標

下載 [ComputeOptimize.yml](#) 檔案。您可以使用此檔案來啟用您帳戶中所有執行個體的記憶體集合。範本檔案會產生下列元件：

- [AWS Systems Manager 參數存放區](#) – 這會存放收集記憶體指標所需的 CloudWatch 代理程式組態。
- AWS Identity and Access Management 連接 [AWS 的 \(IAM\) 角色與受管政策 AWS Systems Manager](#) – 這是 Systems Manager Automation 文件。
- [AWS Systems Manager 文件](#) – 這會安裝和設定 CloudWatch 代理程式（取代任何現有的 CloudWatch 組態）。
- [AWS Systems Manager State Manager 關聯](#) – 這可讓 Systems Manager 文件在您帳戶中的所有執行個體上執行。

### Important

執行此範本會覆寫執行個體上任何現有的 CloudWatch 組態。

接著，執行下列動作：

1. 登入 AWS Management Console 並開啟 [CloudFormation 主控台](#)。
2. 在導覽窗格中，選擇 Stacks (堆疊)。
3. 選擇建立堆疊，然後選擇使用現有資源（匯入資源）。
4. 選擇 Next (下一步)。
5. 針對範本資源，選取上傳範本檔案。
6. 選擇檔案，然後上傳 ComputeOptimize.yml 檔案。
7. 選擇 Next (下一步)。
8. 在指定堆疊詳細資訊頁面上，針對堆疊名稱輸入堆疊的名稱，然後選擇下一步。
9. 在識別資源頁面上，輸入您要匯入之資源的識別符值。
10. 選擇匯入資源。
11. 部署堆疊之後，請選擇輸出索引標籤，以尋找您關聯的金鑰、值和描述。

## 監控關聯進度

1. 部署 CloudFormation 堆疊完成後，請開啟 [Systems Manager 主控台](#)。

2. 在導覽窗格中的節點管理區段中，選擇狀態管理員。
3. 在關聯頁面上，選擇您關聯的關聯 ID。
4. 選擇 Execution history (執行歷程記錄) 標記。
5. 在執行 ID 欄中，選擇您關聯的執行 ID。狀態應為成功。

在 CloudWatch 中檢視指標

建議您等待至少五分鐘，讓指標填入 CloudWatch。

1. 開啟 [CloudWatch 主控台](#)。
2. 在導覽窗格中，展開指標區段，然後選擇所有指標。
3. 確認指標顯示在 CWAgent 命名空間下。

#### Note

若要將設定套用至任何新的執行個體，請重新執行關聯。

## 使用 Compute Optimizer 建議

請考慮一個範例，專注於在單一帳戶和單一區域中進行正確的大小調整。在此範例中，運算最佳化工具會在所有帳戶的組織層級啟用。請記住，適當調整大小是一種破壞性的程序，在大多數情況下，應用程式擁有者會在數週的排程維護時段內以精確度執行。

如果您在組織的管理帳戶中導覽至 Compute Optimizer（如下列步驟所示），您可以選擇要調查的帳戶。在此範例中，在 us-east-1 區域中的單一帳戶中有六個執行中的執行個體。所有六個執行個體都過度佈建。目標是根據 Compute Optimizer 的建議調整執行個體的大小。

識別過度佈建的執行個體和匯出建議詳細資訊

1. 登入 AWS Management Console 並開啟 [Compute Optimizer 主控台](#)。
2. 在導覽窗格中，選擇 Dashboard (儀表板)。
3. 在儀表板頁面上的搜尋方塊中，輸入 Region=US East (N. Virginia)。然後，輸入問題清單=過度佈建。這些篩選條件可讓您查看區域中所有過度佈建的執行個體 us-east-1。
4. 若要檢閱過度佈建 EC2 執行個體的詳細建議，請向下捲動至 EC2 執行個體卡，然後選擇檢視建議。

5. 選擇匯出並儲存檔案以供日後使用。
6. 針對 S3 儲存貯體，輸入您要作為匯出檔案目的地的 Amazon S3 儲存貯體名稱。

#### Note

若要儲存建議以供未來檢閱，您必須有一個 S3 儲存貯體可供 Compute Optimizer 在每個區域中寫入。如需詳細資訊，請參閱 Compute Optimizer 文件中的[適用於的 Amazon S3 儲存貯體政策 AWS Compute Optimizer](#)。

7. 在匯出篩選條件區段中，選取組織中的所有成員帳戶包含建議核取方塊。
8. 針對資源類型，選擇 EC2 執行個體。
9. 在要包含的資料欄區段中，選取全選核取方塊。
10. 選擇 Export (匯出)。

### 根據建議選擇執行個體

執行個體建議是根據 Compute Optimizer 收集和分析的效能指標。請務必注意執行個體上執行的工作負載，以確保您選擇最佳的執行個體。此範例假設您可以從最新一代的 Amazon EC2 [R6i](#)、[R5](#) 和 [T3](#) 執行個體中進行選擇。T3 執行個體可爆量執行，且網路頻寬功能較低。R5 和 R6 執行個體具有相同的每小時成本，且幾乎完全相同。不過，R6 執行個體具有較高的網路頻寬容量、採用最新一代的 Intel 處理器，並提供與 R5 相同的運算使用量。在此範例中，R6 是調整大小的最佳選擇。

1. 在 [Compute Optimizer 主控台](#) 中，從導覽列選擇 EC2 執行個體的建議。此頁面會顯示目前執行個體類型的比較，以及取代它的建議選項。
2. 若要取得您想要正確大小的執行個體 ID，請從管理帳戶開啟 [Amazon S3 主控台](#) AWS Organizations。
3. 在導覽窗格中，選擇儲存貯體，然後選擇您用來存放匯出結果的儲存貯體。
4. 在物件索引標籤上，從物件清單中選取匯出檔案，然後選擇下載。
5. 若要從檔案擷取執行個體資訊，您可以使用 Microsoft Excel 中資料索引標籤上的文字轉資料欄按鈕。

#### Note

執行個體 IDs 以 Amazon Resource Name (ARNs) 表示。請務必將分隔符號設定為 "/"，並擷取執行個體 ID。或者，您可以撰寫指令碼或使用整合開發環境 (IDE) 來修剪 ARN。

6. 在 Excel 中，篩選問題清單欄，以僅顯示 OVER\_PROVISIONED 執行個體。這些是您為適當調整大小而設定目標的執行個體。
7. 將執行個體 IDs 儲存在文字編輯器中，以便稍後輕鬆存取。

## 標記執行個體以正確調整大小

標記工作負載是一種強大的工具，可在中組織您的資源 AWS。標籤可讓您更精細地了解成本，並加速退款。如需將標籤新增至 AWS 資源的策略和方法的詳細資訊，請參閱白皮書標記資源的 AWS 最佳實務。[AWS](#) 在此範例中，您可以使用 [AWS 標籤編輯器](#)，在您想要在維護時段調整大小的過度佈建執行個體之間進行標記調整。您也可以使用此標籤來檢視變更前後的成本。

1. 登入 AWS Management Console 並開啟帳戶的 [AWS Resource Groups 主控台](#)，其中包含用於調整大小的執行個體。
2. 在導覽列的標記區段中，選擇標籤編輯器。
3. 針對區域，選取您的目標區域。
4. 針對資源類型，選擇 AWS::EC2::Instance。
5. 選擇搜尋資源。
6. 在資源搜尋結果頁面上，選取您要正確調整大小的所有執行個體，然後選擇管理所選資源的標籤。
7. 選擇 Add tag (新增標籤)。
8. 針對標籤索引鍵，輸入權利調整。針對標籤值，輸入已啟用。然後，選擇檢閱並套用標籤變更。

### Note

您可以包含其他中繼資料，例如團隊或業務單位，以協助稍後在 Cost Explorer 中篩選。

建立使用者定義的標籤並將其套用至資源後，最多可能需要 24 小時才能在成本分配標籤頁面上顯示標籤，以便啟用。選取標籤進行啟用後，標籤可能需要 24 小時才會變成作用中。

對於進階使用者，您可以在目標帳戶和區域中使用 [AWS CloudShell](#) 來標記多個執行個體。例如：

```
bash
#!/bin/bash
# Set variables
TAG_KEY="rightsizing"
TAG_VALUE="type-m5"
# Get a list of instance IDs
```

```
INSTANCE_IDS=$(aws ec2 describe-instances --query  
  "Reservations[].Instances[].InstanceId" --output text)  
# Loop through each instance ID and add the tag  
for INSTANCE_ID in $INSTANCE_IDS; do  
  aws ec2 create-tags --resources $INSTANCE_ID --tags Key=$TAG_KEY,Value=$TAG_VALUE  
done
```

## 啟用成本分配標籤以使用 AWS 帳單工具

建議您啟用使用者定義的成本分配標籤。這可讓計費 AWS 工具（例如 Cost Explorer 和）中識別和篩選 權利調整標籤 AWS Cost and Usage Report。如果您未啟用此功能，則標籤篩選選項和資料將無法使用。如需使用成本分配標籤的資訊，請參閱 文件中的 AWS 帳單與成本管理 [啟用使用者定義的成本分配標籤](#)。

1. 登入 AWS Management Console 並開啟 [AWS Billing 主控台](#)。
2. 在導覽窗格的帳單區段中，選擇成本分配標籤。
3. 在使用者定義的成本分配標籤索引標籤上，輸入權利調整。
4. 選取調整標籤索引鍵，然後選擇啟用。

24 小時後，標籤應該會出現在 Cost Explorer 中。

## 使用 Systems Manager Automation 實作適當大小的建議

調整大小是一種需要停止和啟動執行個體的案例。在此案例中，您可能需要在維護時段中處理此中斷，並且需要不同的團隊來處理自己的調整規模。變更執行個體類型之前，請檢閱 Amazon EC2 文件中[相容執行個體類型的考量事項](#)。

本節中的範例步驟使用稱為 [AWS-ResizeInstance](#) 的 Systems Manager Automation 文件，實作每個帳戶和區域的正確大小建議。這種方法對於大多數組織來說都是典型的，因為大多數組織需要不同的執行個體類型來實現不同的目的。您也可以使用相同的 AWS-ResizeInstance 自動化文件，以單一和多帳戶部署為目標。

1. 登入 AWS Management Console 並開啟 [Systems Manager 主控台](#)。
2. 在導覽窗格中的共享資源區段中，選擇文件。
3. 在搜尋列中，輸入 AWS-ResizeInstance，然後從搜尋結果中選擇 AWS-ResizeInstance。
4. 選擇 Execute automation (執行自動化)。
5. 在執行自動化執行手冊頁面上，選擇簡易執行。

6. 在輸入參數區段中，輸入 InstanceId 和 InstanceType。保留其餘的預設值。
7. 選擇執行，然後等待自動化完成變更執行個體類型的步驟。

## 考慮替代調整大小方法

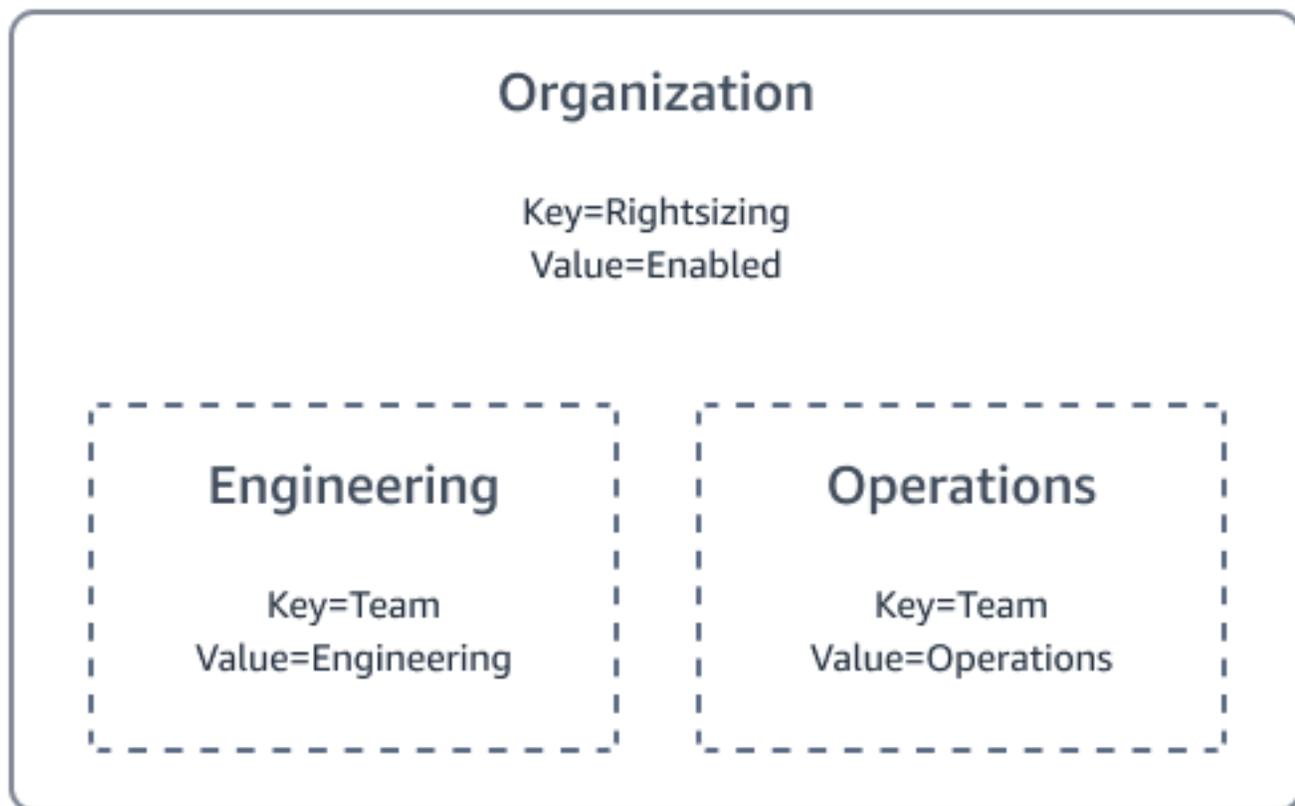
如果您使用啟動範本來部署執行個體，您可以使用適當大小的執行個體類型更新啟動範本，然後執行執行個體重新整理，以適當大小的版本取代執行個體。

如果您打算跨多個帳戶和區域使用正確的大小調整程序，則必須建立自訂 Systems Manager 自動化文件。本文件可讓您在多個執行個體中作為參數饋送，並將目標執行個體移動至相同的目的地執行個體類型（例如，所有執行個體轉換為 t3a.medium，無論來源執行個體類型為何）。

## 在 Cost Explorer 中檢閱成本前後

在您正確調整資源大小之後，您可以使用 Cost Explorer，使用 Rightsizing 標籤來顯示成本前後。請記住，您可以使用[資源標籤](#)來追蹤成本。透過使用多層標籤，您可以精細地了解成本。在本指南中涵蓋的範例中，Rightizing 標籤用於將一般標籤套用至所有目標執行個體。然後，使用團隊標籤來進一步組織資源。下一個步驟是介紹應用程式標籤，以進一步顯示操作特定應用程式的成本影響。

下圖顯示組織的標籤結構。



考慮企業的範例，該企業正確調整營運團隊擁有的生產 Web 伺服器。在 Cost Explorer 中，將 Rightsizing 標籤設定為啟用，並將 Team 標籤設定為操作。在此範例中，正確的調整規模工作可將操作成本從每小時 0.89 美分降低到 0.28 美分。假設每月 744 小時，適當調整規模之前的年成本為 7,945.92 美元。在適當調整規模之後，年成本會降至 2,499.84 美元。這表示年工作負載成本降低了 68.5%。想像一下，這在大型組織中的影響。請記住，這是在範例環境中完成的，而且執行個體大部分是閒置的。在生產環境中，您可以節省 10-35%。

現在，請考慮適當調整工程團隊擁有的生產堡壘主機大小的影響。在 Cost Explorer 中，將 Rightsizing 標籤設定為啟用，並將 Team 標籤設定為工程。在此範例中，適當的調整規模工作會將成本從 0.75 美分降低到每小時 0.44 美分。假設每月 744 小時，適當調整規模之前的年成本為 6,696.00 美元。在適當調整規模之後，年成本會降至 3,928.32 美元。

如果您使用多個標籤，則可以篩選資料，以精細化成本詳細資訊。在此範例中，團隊標籤可減少噪音，讓您可以在團隊層級檢視影響。因為已啟用 權利調整標籤，所以您也可以篩選任何具有該標籤且值為已啟用或不存在值的執行個體。這可以提供正確調整規模工作的整體檢視，特別是在 Cost Explorer 層級的管理帳戶（付款人）中檢視時。此檢視可讓您查看所有帳戶和執行個體。

請考慮單一帳戶層級的範例，其中 Rightsizing 標籤設定為啟用。營運成本從每小時 1.64 美元降至每小時 0.72 美元。假設每月 744 小時，適當調整規模之前的年成本為 14,641.92 美元。適當調整規模後，年成本會降至 6,428.16 美元。這表示此帳戶的運算成本降低了 56%。

開始適當調整大小的旅程之前，請考慮下列事項：

- AWS 提供許多降低成本的選項。這包括 [AWS OLA](#)，其中 會在移至 之前 AWS 檢閱您的現場部署執行個體 AWS。OLA AWS 也為您提供正確的大小調整建議和授權指導。
- 購買 [Savings Plans](#) 之前，請先完成所有正確的大小調整。這可協助您避免在 Savings Plans 承諾上過度購買。

## 建議

我們建議執行下列步驟：

1. 檢閱您現有的環境，並考慮將 Amazon EBS gp2 磁碟區轉換為 gp3 磁碟區。
2. 檢閱 [Savings Plans](#)。

## 其他資源

- [AWS Compute Optimizer](#) (AWS 文件)

- [標記 AWS 資源的最佳實務 \(AWS 白皮書\)](#)
- [如何從 AWS Compute Optimizer 和 AWS Trusted Advisor 跨 \(YouTube\) 收集資料 AWS Organizations YouTube](#)
- [最佳化效能並降低授權成本：利用 AWS Compute Optimizer Amazon EC2 SQL Server 執行個體 \(AWS 部落格上的 Microsoft 工作負載\)](#)

## 為 Windows 工作負載選取正確的執行個體類型

### 概觀

與內部部署環境相比，雲端中操作的工作負載有很大的差異，這是過度佈建的做法。購買內部部署使用的實體硬體時，您預期資本支出會持續一段預定的期間，通常為 3-5 年。為了因應硬體生命週期期間的預期成長，會使用比工作負載目前所需的資源更多的資源來取得硬體。因此，實體硬體通常會過度佈建，遠超過實際工作負載的需求。

虛擬機器 (VM) 技術已成為利用剩餘硬體資源的有效方法。管理員使用 vCPUs 和 RAM 過度佈建的 VMs，讓 Hypervisor 透過將未使用的資源配置給每個 VM，來管理忙碌和閒置伺服器之間的實體資源使用量。管理 VMs 時，配置給每個 VM 的 vCPU 和 RAM 資源會以資源總管的形式運作，而非實際用量的指標。VM 資源過度配置可能會輕鬆超過可用運算資源的三倍。

[Amazon Elastic Compute Cloud \(Amazon EC2\)](#) 不會過度佈建基礎硬體上的 VMs，因為這是不必要的。雲端運算是一種營運費用，而不是資本費用，您只需支付使用的費用。如果您的工作負載未來需要更多資源，請在實際需要時佈建它們，而不是先佔。

選擇正確的 [Amazon EC2 執行個體類型](#) 有數百種選項。如果您打算將 Windows 工作負載遷移至雲端，AWS 會提供 [AWS OLA](#) 協助您更了解目前的工作負載，並提供其在 上效能的範例 AWS。OLA AWS 分析旨在將適當的 EC2 執行個體類型和大小與您的實際現場部署使用量配對。

如果您已有工作負載在 Amazon EC2 上執行並尋求成本最佳化策略，本指南的本節可協助您識別 Amazon EC2 執行個體之間的差異及其對一般 Windows 工作負載的適用性。

### 成本最佳化建議

若要最佳化 EC2 執行個體類型的成本，建議您執行下列動作：

- 為您的工作負載選擇正確的執行個體系列
- 了解處理器架構之間的價格變化
- 了解跨 EC2 世代的效能差異價格

- 遷移至較新的執行個體
- 使用爆量執行個體

## 為您的工作負載選擇正確的執行個體系列

請務必為您的工作負載選擇正確的執行個體系列。

Amazon EC2 執行個體分為下列各種群組：

- 一般用途
- 運算最佳化
- 記憶體最佳化
- 加速運算
- 儲存最佳化
- HPC 最佳化

大多數 Windows 工作負載都屬於下列類別：

- 一般用途
- 運算最佳化
- 記憶體最佳化

若要進一步簡化，請考慮每個類別中的基準 EC2 執行個體：

- 運算最佳化 – C6i
- 一般用途 – M6i
- 記憶體最佳化 – R6i

上一代的 EC2 執行個體在處理器類型中表現出細微差異。例如，C5 運算最佳化執行個體的處理器比 M5 一般用途執行個體或 R5 記憶體最佳化執行個體更快。最新一代的 EC2 執行個體 (C6i、M6i、R6i、C6a、M6a 和 R6a) 都會跨執行個體系列使用相同的處理器。由於處理器在最新一代的執行個體之間是一致的，因此執行個體系列之間的價格差異現在更取決於 RAM 的數量。執行個體擁有的 RAM 越多，成本就越高。

下列範例說明在 us-east-1 區域中執行之 Intel 4 vCPU 執行個體的每小時定價。

| 執行個體       | vCPU | RAM | 每小時價格   |
|------------|------|-----|---------|
| c6i.xlarge | 4    | 8   | 0.17 美元 |
| m6i.xlarge | 4    | 16  | 0.19 美元 |
| r6i.xlarge | 4    | 32  | 0.25 美元 |

### Note

定價是以 us-east-1 區域中的隨需每小時定價為基礎。

## 爆量執行個體

雖然關閉未使用的運算資源以避免產生費用是雲端運算的最佳實務，但並非所有工作負載都可以在每次需要時關閉和開啟。有些工作負載會長時間保持閒置，但必須每天 24 小時都能存取。

爆量執行個體 (T3) 提供一種方法，讓您整天維持線上尖峰或低使用率工作負載，同時保持低運算成本。爆量 EC2 執行個體具有執行個體可在短時間內使用的 vCPU 資源數量上限。這些執行個體採用以 [高載 CPU 點數](#) 為基礎的系統。這些點數會在一天的閒置期間累積。爆量執行個體提供不同的 vCPU-to-RAM 比率，讓它們成為在某些情況下運算最佳化執行個體，以及在其他情況下其他一般用途執行個體的替代方案。

下列範例說明 us-east-1 區域中執行的 T3 執行個體（即爆量執行個體）的每小時定價。

| 執行個體      | vCPU | RAM (GB) | 每小時價格     |
|-----------|------|----------|-----------|
| t3.nano   | 2    | 0.5      | 0.0052 美元 |
| t3.micro  | 2    | 1        | 0.0104 美元 |
| t3.small  | 2    | 2        | 0.0208 美元 |
| t3.medium | 2    | 4        | 0.0416 美元 |
| t3.large  | 2    | 8        | 0.0832 美元 |
| t3.xlarge | 4    | 16       | 0.1664 美元 |

| 執行個體       | vCPU | RAM (GB) | 每小時價格     |
|------------|------|----------|-----------|
| t3.2xlarge | 8    | 32       | 0.3328 美元 |

**Note**

定價是以 us-east-1 區域中的隨需每小時定價為基礎。

## 了解處理器架構之間的價格變化

[Intel](#) 處理器自成立以來即為 EC2 執行個體的標準。較早世代的 EC2 執行個體，例如 C5, M5 和 R5，不會指出 Intel 做為處理器架構（因為這是預設值）。較新一代的 EC2 執行個體，例如 C6i, M6i 和 R6i，包含表示 Intel 處理器使用的「i」。

處理器架構註釋的變更是由於引入了其他處理器選項。與 Intel 最相似的處理器是 [AMD](#)（以「a」表示）。AMD EPYC 處理器使用相同的 x86 架構，並提供類似於 Intel 處理器的效能，但價格較低。如下列定價範例所示，AMD EC2 執行個體的運算成本相較於 Intel 執行個體提供大約 10% 的折扣。

| Intel 執行個體 | 每小時價格    | AMD 執行個體   | 價格        | % 差異 |
|------------|----------|------------|-----------|------|
| c6i.xlarge | 0.17 美元  | c6a.xlarge | 0.153 美元  | 10%  |
| m6i.xlarge | 0.192 美元 | m6a.xlarge | 0.1728 美元 | 10%  |
| r6i.xlarge | 0.252 美元 | r6a.xlarge | 0.2268 美元 | 10%  |

**Note**

定價是以 us-east-1 區域中的隨需每小時定價為基礎。

第三個主要處理器架構選項是 EC2 執行個體上的 [AWS Graviton 處理器](#)（以 "g" 表示）。Graviton 處理器由設計 AWS，可在 Amazon EC2 上提供最佳價格效能。目前 Graviton 處理器不僅比 Intel 處理器便宜 20%，也提供 20% 或更高的效能提升。新一代的 Graviton 處理器預期會進一步擴展此效能差異，而測試顯示效能額外增加 25%。

Windows Server 無法在以 ARM 架構為基礎的 Graviton 處理器上執行。事實上，Windows Server 只會在 x86 處理器上執行。雖然您無法透過使用 Graviton 型 Windows Server 執行個體來提高 40% 的價格效能，但您仍可以搭配特定 Microsoft 工作負載使用 Graviton 處理器。例如，[較新版本的 .NET 可以在 Linux 上執行](#)。這表示這些工作負載可以使用 ARM 處理器，並受益於更快、更經濟實惠的 Graviton EC2 執行個體。

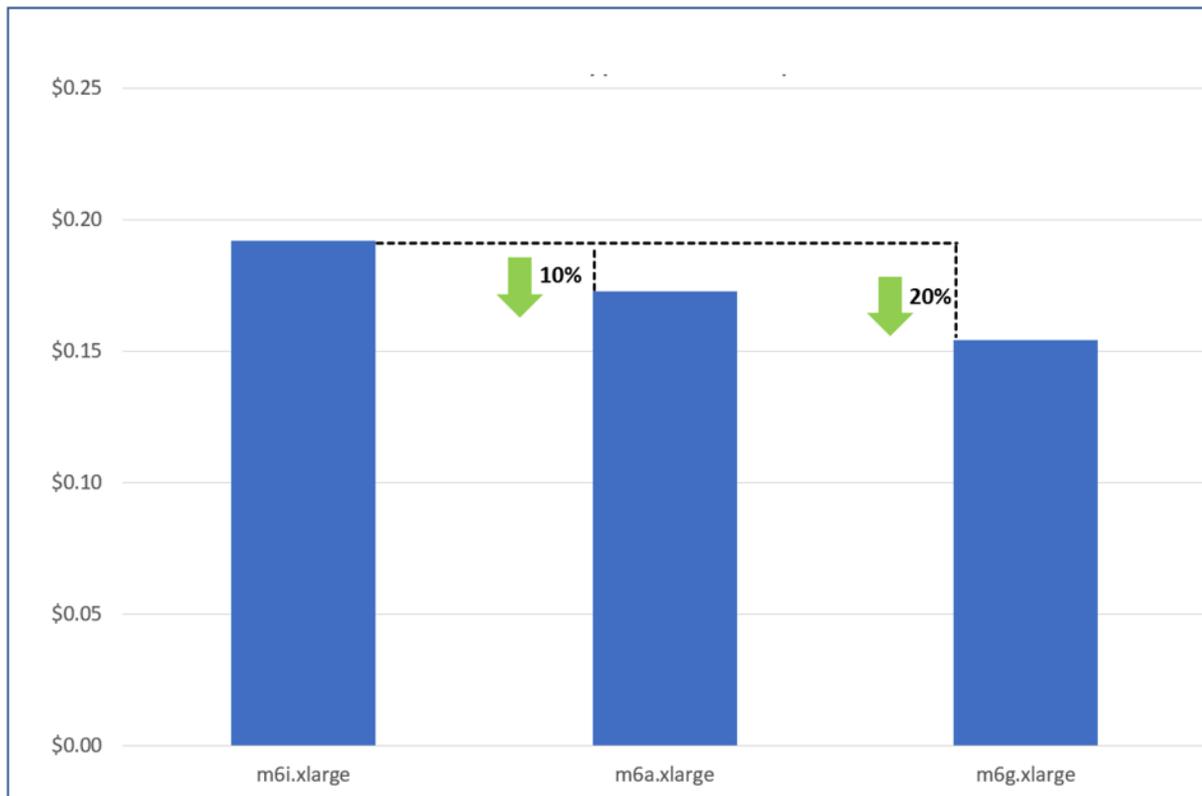
下列範例說明 us-east-1 區域中執行的 Graviton 執行個體的每小時定價。

| Intel 執行個體 | 每小時價格    | Graviton 執行個體 | 每小時價格     | % 差異 |
|------------|----------|---------------|-----------|------|
| c6i.xlarge | 0.17 美元  | c6g.xlarge    | 0.136 美元  | 20%  |
| m6i.xlarge | 0.192 美元 | m6g.xlarge    | 0.154 美元  | 20%  |
| r6i.xlarge | 0.252 美元 | r6g.xlarge    | 0.2016 美元 | 20%  |

 Note

定價是以 us-east-1 區域中的隨需每小時定價為基礎。

下圖比較 M 系列執行個體的價格。



## 了解跨 EC2 世代的價格效能差異

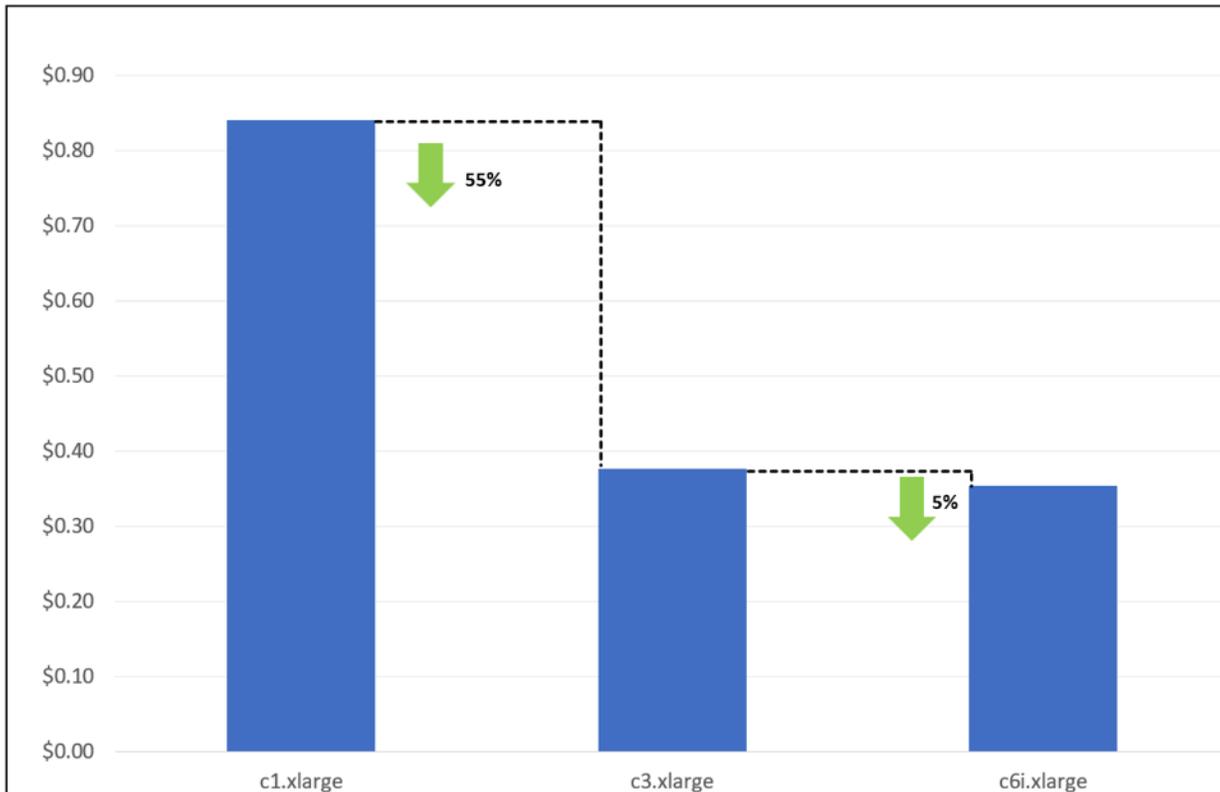
Amazon EC2 最一致的特性之一是，每個新一代都比上一代產品提供更好的價格效能。如下表所示，較新一代 EC2 執行個體的價格會隨後續版本而降低。

| 運算最佳化執行個體 | 每小時價格   | 一般用途執行個體  | 每小時價格    | 記憶體最佳化執行個體 | 每小時價格    |
|-----------|---------|-----------|----------|------------|----------|
| C1.xlarge | 0.52 美元 | M1.xlarge | 0.35 美元  | r1.xlarge  | N/A      |
| C3.xlarge | 0.21 美元 | M3.xlarge | 0.266 美元 | r3.xlarge  | 0.333 美元 |
| C5.xlarge | 0.17 美元 | M5.xlarge | 0.192 美元 | r5.xlarge  | 0.252 美元 |

### Note

定價是以 us-east-1 區域中的隨需每小時定價為基礎。

下表比較不同世代 C 系列執行個體的成本。



不過，第 6 代執行個體的价格與第 5 代相同，如下表所示。

| 運算最佳化執行個體  | 每小時價格   | 一般用途執行個體   | 每小時價格    | 記憶體最佳化執行個體 | 每小時價格    |
|------------|---------|------------|----------|------------|----------|
| C5.xlarge  | 0.17 美元 | M5.xlarge  | 0.192 美元 | r5.xlarge  | 0.252 美元 |
| C6i.xlarge | 0.17 美元 | M6i.xlarge | 0.192 美元 | r6i.xlarge | 0.252 美元 |

#### Note

定價是以 us-east-1 區域中的隨需每小時定價為基礎。

雖然成本相同，但由於處理器速度更快、聯網輸送量增強，以及 Amazon Elastic Block Store (Amazon EBS) 輸送量和 IOPS 增加，新一代提供了卓越的價格效能。

其中一個最重要的價格效能改進是 [X2i 執行個體](#) 的增強功能。此世代的執行個體提供比上一代多 55% 的價格效能。如下表所示，x2iedn 會在每個效能方面（全部與上一代的價格相同）展現改善。

| 執行個體           | 每小時價格   | vCPU | RAM | 處理器速度   | 執行個體儲存體        | 聯網      | Amazon EBS 輸送量 | EBS IOPS |
|----------------|---------|------|-----|---------|----------------|---------|----------------|----------|
| x1e.2xlarge    | 1.66 美元 | 8    | 244 | 2.3 GHz | 237GB SSD      | 10 Gbps | 125 MB/s       | 7400     |
| x1iedn.2xlarge | 1.66 美元 | 8    | 256 | 3.5 GHz | 240GB NVMe SSD | 25 Gbps | 2500 MB/s      | 65000    |

### Note

定價是以 us-east-1 區域中的隨需每小時定價為基礎。

## 範例方案

請考慮追蹤交付車輛並希望改善 SQL Server 效能的分析公司範例。MACO SME 審查公司的效能瓶頸後，公司會從 x1e.2xlarge 執行個體轉換為 x2iedn.xlarge 執行個體。新的執行個體大小較小，但 x2 執行個體的增強功能可透過使用緩衝集區延伸來提高 SQL Server 效能和最佳化。這可讓公司從 SQL Server Enterprise Edition 降級為 SQL Server Standard Edition。它還使公司能夠將其 SQL Server 授權從 8 個 vCPUs 減少到 4 vCPUs。

最佳化之前：

| Server  | EC2 執行個體    | SQL Server 版本 | 每月成本        |
|---------|-------------|---------------|-------------|
| ProdDB1 | x1e.2xlarge | Enterprise    | 3,918.64 美元 |
| ProdDB2 | x1e.2xlarge | Enterprise    | 3,918.64 美元 |
| 總計      |             |               | 7,837.28 美元 |

最佳化後：

| Server  | EC2 執行個體      | SQL Server 版本 | 每月成本        |
|---------|---------------|---------------|-------------|
| ProdDB1 | x2iedn.xlarge | 標準            | 1,215.00 美元 |
| ProdDB2 | x2iedn.xlarge | 標準            | 1,215.00 美元 |
| 總計      |               |               | 2,430.00 美元 |

所有合併後，從 x1e.2xlarge 執行個體變更為 x2iedn.xlarge 執行個體，可讓範例案例中的公司每月在生產資料庫伺服器上節省 5,407 美元。這可將工作負載的總成本降低 69%。

#### Note

定價是以 us-east-1 區域中的隨需每小時定價為基礎。

## 遷移至較新的執行個體

較舊世代的 Amazon EC2 會在 Xen Hypervisor 上執行，而較新世代的 Amazon EC2 會在 [AWS Nitro 系統](#) 上執行。Nitro 系統可將主機硬體的幾乎所有運算和記憶體資源交付給您的執行個體。這可改善整體效能。[從 Xen 遷移到 Nitro 型執行個體](#) 時有特殊考量。例如，[AWS Windows AMIs](#) 是使用預設設定和 Microsoft 安裝媒體使用的自訂來設定。自訂包含支援最新一代執行個體類型的驅動程式和組態 ([在 Nitro 系統上建置的執行個體](#))。

如果您要從自訂 Windows AMIs 或從 Amazon 提供的 Windows AMIs 啟動執行個體，而這些執行個體是在 2018 年 8 月之前建立，建議您完成從 [遷移到 Amazon EC2 文件中最新一代執行個體類型的步驟](#)。Amazon EC2

## 使用爆量執行個體

雖然爆量執行個體是節省運算成本的好方法，但建議您在下列情況下避免這些執行個體：

- 具有桌面體驗的 [Windows Server 最低規格](#) 需要 2 GB 的 RAM。避免將 t3.micro 或 t3.nano 執行個體與 Windows Server 搭配使用，因為它們缺少最低數量的 RAM。
- 如果您的工作負載尖峰，但閒置時間不夠長，無法建置爆量額度，則使用一般 EC2 執行個體比使用爆量執行個體更有效率。建議您 [監控 CPU 點數](#)，以驗證這一點。

- 在大多數情況下，建議您避免將爆量執行個體與 SQL Server 搭配使用。SQL Server 的授權是根據指派給執行個體 vCPUs 數量。如果 SQL Server 在一天的大部分時間都處於閒置狀態，則您需要為未充分利用的 SQL 授權付費。在這些情況下，我們建議您將多個 SQL Server 執行個體合併到更大的伺服器上。

## 後續步驟

建議您採取下列後續步驟，以最佳化 Amazon EC2 Windows 執行個體的成本：

- 使用最新一代的 EC2 執行個體以獲得最佳價格效能。
- 搭配 AMD 處理器使用 EC2 執行個體，可降低 10% 的運算成本。
- 選擇符合您工作負載的 EC2 執行個體類型，將資源使用率最大化。

下表顯示 Windows 工作負載的一般起點範例。還有其他選項可用，例如執行個體儲存磁碟區，可增強 SQL Server 工作負載，或具有更大 vCPU-to-RAM 比率的 EC2 執行個體。建議您徹底測試工作負載，並使用等監控工具 AWS Compute Optimizer 來協助進行必要的調整。

| 工作負載             | 典型      | 選用            |
|------------------|---------|---------------|
| Active Directory | T3, M6i | R6i           |
| 檔案伺服器            | T3, M6i | C6i           |
| Web 伺服器          | T3, C6i | M6i, R6i      |
| SQL Server       | R6i     | x2iedn、X2iezn |

如果您必須變更 EC2 執行個體類型，程序通常只需要簡單的伺服器重新啟動。如需詳細資訊，請參閱 [Amazon EC2 文件中的變更執行個體類型](#)。Amazon EC2

在您變更執行個體類型之前，建議您考慮下列事項：

- 您必須先停止 Amazon EBS 支援的執行個體，才能變更其執行個體類型。請務必在執行個體停止時規劃停機時間。停止執行個體並變更其執行個體類型可能會花費數分鐘，重新啟動您的執行個體所需要花費的時間則會根據您應用程式的啟動指令碼而有所不同。如需詳細資訊，請參閱 Amazon EC2 文件中的 [停止和啟動執行個體](#)。

- 當您停止和啟動執行個體時，會將執行個體 AWS 移至新的硬體。如果您的執行個體具有公有 IPv4 地址，會 AWS 釋出該地址，並為執行個體提供新的公有 IPv4 地址。如果您需要未變更的公有 IPv4 地址，請使用[彈性 IP 地址](#)。
- 如果在執行個體上啟用[休眠](#)，則無法變更執行個體類型。
- 您無法變更 [Spot 執行個體](#)的執行個體類型。
- 如果您的執行個體位於 Auto Scaling 群組中，Amazon EC2 Auto Scaling 會將停止的執行個體標記為運作狀態不佳，並可能會將其終止並啟動替代執行個體。若要避免這種情況，您可以在變更執行個體類型時，暫停群組的擴展程序。如需詳細資訊，請參閱 [Amazon EC2 Auto Scaling 文件中的暫停和繼續 Auto Scaling 群組的程序](#)。Amazon EC2 Auto Scaling
- 當您變更具有 NVMe 執行個體存放區磁碟區的執行個體類型時，更新的執行個體可能會有額外的執行個體存放區磁碟區，因為即使未在 Amazon Machine Image (AMI) 或執行個體區塊型儲存設備映射中指定，所有 NVMe 執行個體存放區磁碟區都可以使用。否則，更新後的執行個體通常會有與您在啟動原始執行個體時指定之執行個體儲存體磁碟區相同數目的執行個體儲存體磁碟區。

## 其他資源

- [Amazon EC2 執行個體類型](#) (AWS 文件)
- [AWS 最佳化和授權評估](#) (AWS 文件)

## 為 Windows 和 SQL Server 工作負載提供授權

### 概觀

如果您對 Microsoft 工作負載和現有的企業授權合約有大量投資，您可以選擇多種 AWS 選項來支援這些工作負載，包括[包含的授權 \(由提供 AWS\)](#) 和 [自帶授權 \(BYOL\)](#) 選項。您可以使用 [Amazon EC2 專用主機](#) 來充分利用現有的 Microsoft 授權合約，並將 Windows Server 帶入其中 AWS。這可為您節省高達 50% 的 Amazon EC2 執行個體成本。由於 Windows 授權約佔執行個體成本的一半，因此將 Windows Server 帶到專用主機 AWS 上的，可大幅節省成本。由於 Windows Server 無法設為[預設 \(共用\) 租用](#)，因此如果您想要使用現有的 Windows Server 授權，專用主機是理想的選擇 AWS。

專用主機不僅適用於 Windows Server BYOL 執行個體。它們也可讓您靈活地比對現有 SQL Server 工作負載的內部部署授權。專用主機公開基礎伺服器的實體核心，並可讓您在實體核心層級授權 SQL Server。預設 (共用) 租用中，SQL Server 授權是根據配置給執行個體 CPUs 數目而定的，這是不可能的。此功能可讓您在 AWS 以與您的內部部署授權策略一致的方式授權 SQL Server 工作負載。因此，相較於預設 (共用) 租用，除了節省執行個體成本之外，您還可以使用合格的 Windows 授權，

節省高達 50% 的 SQL Server 授權成本。如需此案例的詳細資訊，請參閱本指南的[了解 SQL Server 授權](#)一節。

## Amazon EC2 專用執行個體

Amazon EC2 專用主機基本上與 AWS 用來執行其 EC2 運算產品的 EC2 主機相同。差別在於，這些主機完全專用於單一客戶，並提供對基礎實體基礎設施的專屬存取權。您可以使用專用主機在完全專用於您的硬體上執行執行個體，而不是與其他 AWS 客戶共用資源。這可讓您更妥善地控制雲端資源，並讓您透過將 Windows Server 和 SQL Server 等自有軟體授權帶入其中來降低成本 AWS。

請謹記以下幾點：

- 專用主機是完全專用於單一客戶的實體伺服器。您可以查看專用主機的通訊端和實體核心，以便解決授權合規要求，例如每個通訊端、每個核心或每個虛擬機器軟體授權合約。
- 可支援相同執行個體系列的多個執行個體大小的專用主機稱為異質專用主機。這些[執行個體系列](#)包括 T3, A1, C5, M5, R5, C5n, R5n 和 M5n。相反地，其他執行個體系列僅支援相同專用主機上的一個執行個體大小。這些稱為同質專用主機。
- 專用主機是以每個主機為基礎計費。這表示無論您在專用主機上執行多少執行個體，都會向您收取每個專用主機的費用。專用主機定價會根據選取的執行個體系列、區域和付款選項而有所不同。您可以為工作負載選擇最佳組態，以實現所需的效能和成本成果。

此圖說明共用租用執行個體與專用主機之間的差異。



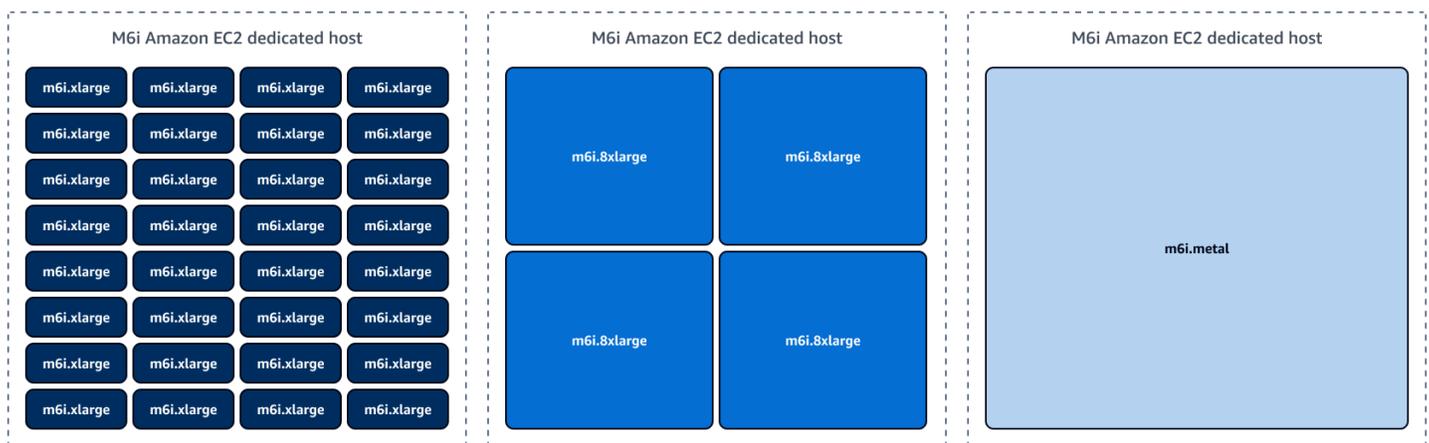
## 同質專用主機

考慮使用 M6i 專用主機的情況。M6i 和 R6i 專用主機有兩個通訊端、64 個實體核心，並支援相同大小的執行個體類型。這些稱為同質專用主機。這表示您可以在單一 M6i 專用主機上啟動的執行個體數量取決於執行個體大小。

例如：

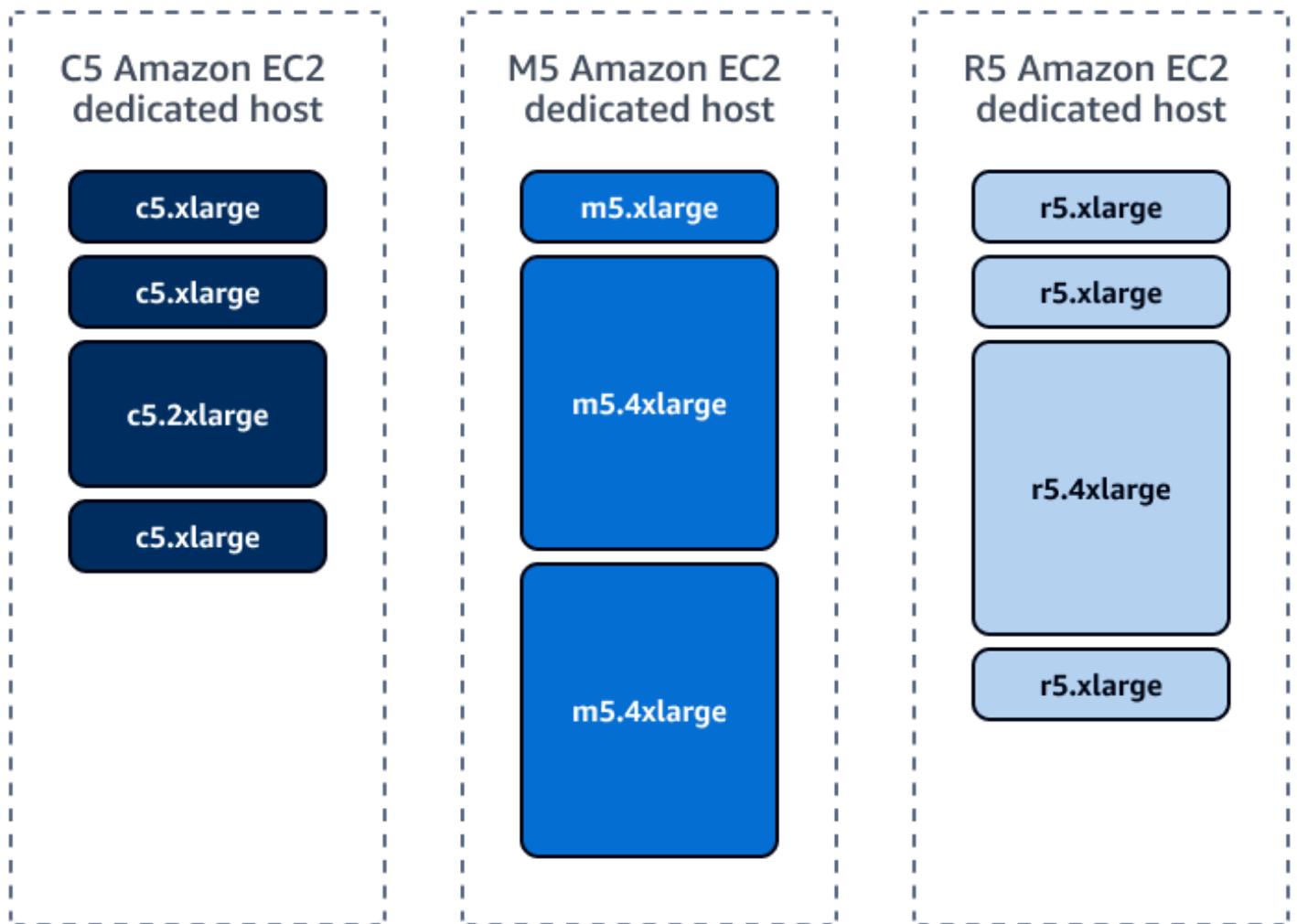
- 若是 xlarge (4 vCPUs)，您最多可以在此專用主機上啟動 32 個 m6i.xlarge 執行個體。
- 如果是 8xlarge (32 個 vCPUs)，您最多可以在此專用主機上啟動 4 個 m6i.8xlarge 執行個體。
- 如果是金屬 (128 個 vCPUs)，您最多可以在此專用主機上啟動 1 個 m6i.metal 執行個體。

下圖顯示 M6 執行個體的專用主機選項。



## 異質專用主機

支援相同主機上多個執行個體大小的專用主機稱為異質 Amazon EC2 專用主機。下圖顯示具有各種執行個體大小的 C5, M5 和 R5 專用主機範例，例如 2xlarge、xlarge 和 4xlarge。



## 專用主機管理

我們建議您在管理 Amazon EC2 專用主機時考慮下列事項：

- 若要充分利用專用主機，您可以在[組織內的多個帳戶之間共用單一主機](#)。主機共用可讓資源最佳化，並可透過使用主機上的每個可用插槽來節省成本。透過在業務單位之間共用專用主機，您可以集中 IT 基礎設施並改善資源使用率，同時保持工作負載之間的區隔。如果您屬於中的組織，AWS Organizations 並在組織內啟用共用，則組織中的消費者會自動獲得共用專用主機的存取權。否則，消費者會收到加入資源共享的邀請，並且在接受邀請後便能存取共用的專用執行個體。
- 您可以在包含授權的模型下的專用主機上執行 Windows Server 2022，因為 Windows Server 2019 是您可以 BYOL 的最新版本。如果您想要在專用主機上使用 Windows Server 2022，則必須使用 Windows Server 2022 授權包含的執行個體。
- [AWS License Manager](#) 是全方位解決方案，可管理來自跨 AWS 和內部部署環境之各種廠商的軟體授權。如果您[使用 License Manager](#)，您可以更清楚地了解和控制軟體授權的使用方式，進而節省成

本並改善合規性。您可以使用 License Manager 設定規則來模擬您的唯一授權條款。這可讓您強制執行這些規則，並防止授權濫用。這可以降低不合規的風險，並改善授權管理程序。

- 您可以使用 License Manager，透過使用主機[資源群組](#)來自動化主機的置放、發行和復原。這可以提高生產力並減少管理開銷。License Manager 也會根據授權規則，集中檢視跨 AWS 內部部署環境的授權用量，讓您輕鬆地管理整個組織的增量授權購買、合規和廠商稽核。此外，License Manager 會與 AWS Organizations 和 AWS Resource Access Manager (AWS RAM) 整合，以跨帳戶和區域共用授權組態。這可讓您根據排程建立整個環境的報告，並集中管理授權規則 AWS 帳戶。最後，這可以改善控管並降低複雜性。
- 為單一區域內的專用主機設計高可用性時，請確定您已在生產關鍵工作負載的至少兩個可用區域中配置至少兩個專用主機。如需詳細資訊，請參閱參考部署上的 [Amazon EC2 Microsoft Windows 專用主機 AWS](#)。
- 對於每個專用主機執行個體系列，您可以為每個執行個體大小執行的執行個體數量都有限制。如需詳細資訊，請參閱 Amazon EC2 文件中的[專用主機組態表](#)。

## AWS 授權選項

授權分為下列主要類別：

- 包含授權 – 此授權選項可讓您隨需購買和使用授權，只需支付使用量的費用。它非常適合您在授權使用上尋求靈活性，並希望避免預付成本的使用案例。您可以從各種 Windows Server、SQL Server 和其他 Microsoft 產品中進行選擇。
- 具有授權行動性的 BYOL 產品 – 如果您已經有現有的授權，並想要在雲端中使用它們，此授權選項可讓您透過 [Microsoft 授權行動性計劃](#) 將自己的授權帶到雲端。具有授權行動性的產品，例如具有軟體保證 (SA) 的 SQL Server，可以用於共用或專用租用。這可降低 AWS 執行個體成本。
- 沒有授權行動性的 BYOL 產品 – 對於 Windows Server 這類缺少授權行動性的 Microsoft 產品，AWS 提供在雲端使用這些產品的專用選項。此外，專用主機可在實體核心層級啟用授權，因此可節省 50% 或更多執行工作負載所需的授權。專用主機是大多數時間執行的穩定且可預測工作負載的絕佳選擇。

## 使用 Windows Server 授權

使用您自己的 Windows 授權是授權最佳化最有效的策略之一，因為它可讓您利用現有的投資並降低您的 AWS 費用。特定 BYOL 案例不需要 SA 或授權行動性利益，但 Amazon EC2 專用基礎設施始終是必要的。若要符合資格，您必須在 2019 年 10 月 1 日之前購買永久授權，或在 2019 年 10 月 1 日之前根據有效的 Enterprise 註冊新增永久授權做為校正。在這些特定的 BYOL 案例中，您只能將授

權升級至 2019 年 10 月 1 日之前可用的版本。例如，如果您在 2017 年捨棄 SA，則您有權僅部署至 Windows Server 2016，而非 2019。不過，2019 年是符合 BYOL 資格的最後一個版本 AWS。如需詳細資訊，請參閱 AWS 文件中的[授權 – Windows Server](#)。

使用授權可能會大幅影響執行 Microsoft 工作負載的成本 AWS。當您使用自己的授權時，您不需要為在雲端中執行的執行個體支付額外的授權成本，這可以大幅節省成本。

下表顯示在各種組態上全年無休執行單一 c5.xlarge 執行個體的隨需每月成本。

| 組態                                       | 每月成本 ( 美元 )      |
|--|------------------|
| Windows Server + SQL Server Enterprise 版 | 1,353.00 美元 (LI) |
| Windows Server + SQL Server Standard 版   | 609.00 美元 (LI)   |
| 僅限 Windows Server                        | 259.00 美元 (LI)   |
| 僅限運算 (Linux)                             | 127.00 美元        |

您可以使用現有的授權來降低授權成本，並節省整體 AWS 帳單的費用。

若要符合 Amazon EC2 專用主機上的 BYOL 資格，您必須自備軟體授權，例如 for Windows Server 和 SQL Server。BYOL 可讓您在 上使用現有的授權 AWS，並節省成本。若要自帶授權，您必須擁有軟體廠商的授權權利，也必須提供軟體的安裝媒體或映像。安裝媒體或映像可用於在專用主機上啟動執行個體。若要進一步了解如何建立 BYOL AMI，請參閱 AWS 部落格上的 Microsoft 工作負載中的[如何使用 VM Import/Export 從內部部署建立 Windows Server Bring-Your-Own-License AMIs](#)。

#### Note

設定為 Auto 的授權類型相當於[AWS 包含授權的選項](#)。此選項可能會導致不需要的隨需支出。您需要切換[授權類型](#)。

## 成本最佳化案例

正確調整和最佳化授權是 上成本最佳化的關鍵元件 AWS。如果您實作正確的策略，則可以使用 Amazon EC2 專用主機和 BYOL 選項，降低授權成本、維持合規，並從授權投資中達到最佳價值。

本節涵蓋下列範例案例：

- 使用 T3 專用主機節省成本
- 使用 SQL Server BYOL 比較共用租用與專用主機
- 高度可用的 SQL Server 部署

## 使用 T3 專用主機節省成本

T3 專用主機不同於傳統上提供固定 CPU 資源的其他 Amazon EC2 專用主機。相反地，T3 專用主機支援爆量執行個體，這些執行個體能夠共用 CPU 資源、提供基準 CPU 效能，以及在需要時爆量。共享 CPU 資源也稱為過度訂閱，這可讓單一 T3 專用主機支援比同等一般用途專用主機多四倍的執行個體。

T3 專用主機透過提供比任何其他 Amazon EC2 專用主機更高的執行個體密度來提高 TCO。爆量 T3 執行個體可讓您在比以往更少的主機上，將更多執行個體合併為 low-to-moderate CPU 使用率。與其他 Amazon EC2 專用主機相比，T3 專用主機也在更多 vCPU 和記憶體組合中提供較小的執行個體大小。較小的執行個體大小有助於降低 TCO，並協助提供相當於或大於內部部署主機的合併比率。

T3 專用主機最適合以 low-to-moderate 使用率執行 BYOL 軟體，且每個通訊端、每個核心或每個虛擬機器軟體授權都符合資格，包括 Microsoft Windows 桌面、Windows Server、SQL Server 和 Oracle Databases。

## 使用 T3 專用主機來減少 Windows Server Datacenter 授權（每個核心）

在內部部署環境中，您會利用一個事實：您可以輕鬆地在 VMware 主機上過度訂閱實體 CPUs，並實現高度整合。

請考量下列範例。您目前在內部部署環境中使用 10x36 核心、384 GB RAM VMware 主機。此外，每個主機執行 96x2 vCPU、4 GB RAM Windows Server 虛擬機器，且 CPU 使用率平均值較低。

您現在可以透過將虛擬機器移至 T3 專用主機來達成更高層級的整合，其 RAM 數量是目前現場部署 VMware 主機的兩倍。您可以在 T3 專用主機上執行相同數量的伺服器，主機成本降低 50%。這可協助您將 Windows Server 授權成本降低 33%。下表重點介紹使用 T3 專用主機的節省成本。

|           | 內部部署 VMware 主機 | T3 專用主機 | 節省 |
|-----------|----------------|---------|----|
| 實體伺服器     | 10             | 5       |    |
| 每個主機的實體核心 | 36             | 48      |    |

|  | 內部部署 VMware 主機 | T3 專用主機      | 節省  |
|--|----------------|--------------|-----|
| 每個主機的 RAM (GB)   | 384            | 768          |     |
| 2 個 vCPU，每個主機<br>4 GB RAM VMs                                  | 96             | 192          |     |
| VMs 總數   | 960            | 960          |     |
| Windows Server<br>Datacenter 授權總數<br>(每個核心) = (伺服器數目 * 實體核心計數) | 10 * 36 = 360  | 5 * 48 = 240 | 33% |

## 使用 SQL Server BYOL 比較共用租用與專用主機

請考慮實際範例，以示範 Amazon EC2 專用主機的值。在此案例中，組織在具有 240 個核心的內部部署環境中執行 SQL Server 工作負載，並希望以具成本效益的方式部署相同的工作負載 AWS。如果此組織擁有自己的授權 (BYOL)，他們將繼續支付 SA 的費用，並減少核心數量直接影響其成本。

下圖比較 Microsoft 權利和 SQL Server 之間的 AWS 節省。

| Microsoft entitlements (Enterprise Agreements) |                  | SQL Server savings with AWS |                                |
|--|------------------|-----------------------------|--------------------------------|
|  | Number of cores  | AWS shared vCPUs            | AWS BYOL/Dedicated Hosts cores |
| SQL Server Enterprise edition                  | 208              | 120                         | 96                             |
| SQL Server Standard edition                    | 32               | 20                          | -                              |
| <b>Total SA cost</b>                           | <b>\$341,000</b> | <b>\$197,418</b>            | <b>\$151,355</b>               |

透過在 AWS 共用租用上正確調整執行個體大小，您可以將 SQL Server 授權減少至 140 個核心。這會產生 197,000 美元的 SA 成本。

Amazon EC2 專用主機可讓您在實體核心層級授權 SQL Server。在 SQL Server 授權以配置給執行個體的 vCPUs 數量為基礎的共用租用中，這是不可能的。因此，透過使用兩個 R5 專用主機搭配每個 48 個核心，您只需要涵蓋 96 個核心，而不是共用租用所需的 140 vCPUs。透過在實體層級部署 R5 專用主機和授權工作負載，您可以將所需的 SQL Server Enterprise Edition 授權數量縮減至 96 個核心。這表示您可以部署多達 192 個 SQL Server 工作負載的核心（考量超執行緒），同時仍符合授權需求並大幅節省成本。

在此情況下，組織每年支付約 341,000 USD 的 SA 成本。在適當調整共用租用的大小之後，它們會使用 140 個 vCPUs 將成本降低至 197,000 美元。Amazon EC2 專用主機進一步將成本降低至 151,000 美元（約減少 56%）。

## 高度可用的 SQL Server 部署

此範例會分析成本如何影響上的 AWS SQL Server 部署，並考量各種授權。假設組織需要在上部署六個 SQL Server Enterprise 伺服器 AWS，以支援三個應用程式。這些伺服器需要高可用性，每個伺服器都有 16 vCPUs 和 256 GB 的 RAM。請參閱下列案例詳細資訊：

- 伺服器 – SQL Server
- 作業系統版本 – Windows Server Datacenter 2019
- SQL Server 版本 – SQL Server Enterprise 2019
- vCPU – 16
- 記憶體 (GB) – 256
- 數量 – 6

若要在不影響效能 AWS 的情況下最佳化上的成本，建議您根據 CPU、記憶體、網路和磁碟 (IOPS/BW) 使用率來調整執行個體大小。正確調整工作負載大小後，請將它們放在 x2iedn.4xlarge 執行個體類型上，該類型提供 16 vCPUs。不過，此執行個體類型也包含工作負載所需記憶體的兩倍。仍然可以進一步最佳化。

### 案例 1

組織使用 Windows 和 SQL Server 的包含授權選項，在 AWS 共用租用上部署六個 SQL Server Enterprise 伺服器。使用此選項，Windows 和 SQL Server 授權的成本會併入執行個體價格。請參閱下列案例詳細資訊：

- 共用租用（執行個體） – x2iedn.4xlarge
- 每小時成本 (USD) – 10.0705 美元
- 每月每單位成本 (USD) – 7,351.47 美元
- 伺服器數量 – 6
- CPU – 16
- 記憶體 – 512
- 6 部伺服器的每月成本 – 44,108 美元

## 案例 2

一個組織在共用租用上有適用於 SQL Server 的 SA 和 BYOL。這表示組織使用 Windows 的包含授權選項，但會根據配置給執行個體 vCPUs 數目提供自己的 SQL Server 授權。由於組織有六個 SQL Server Enterprise 伺服器，每個伺服器有 16 vCPUs，因此總共需要 96 vCPUs。請參閱下列案例詳細資訊：

- 共用租用（執行個體）– x2iedn.4xlarge
- 每小時成本 (USD) – 4.0705 美元
- 每月每單位成本 (USD) – 2971.47 美元
- 伺服器數量 – 6
- CPU – 16
- 記憶體 – 512
- BYOL 核心 – 96
- 6 部伺服器的每月成本 – 17,828 美元

透過使用 SA 取得自己的 SQL Server 授權，此案例中的組織相較於使用 SQL Server 的包含授權選項，可以節省成本。精確的成本節省取決於特定授權合約的定價和條款。在此案例中，將 SQL Server Enterprise 授權提供給時，AWS 成本每月會降低 26,280 美元 AWS。

## 案例 3

組織在 Amazon EC2 專用主機上同時具有適用於 Windows 和 SQL Server 的 BYOL。這表示組織將在實體核心層級指派授權，讓他們只能授權主機的實體核心。在實體核心層級的授權可讓您部署執行個體數量上限，而不會影響所需的授權。此授權模型通常與 Windows Server Datacenter 和 SQL Server Enterprise Edition 搭配使用。

此案例使用兩個 X2iezn Amazon EC2 專用主機。每個主機都有 24 個實體核心和 48 vCPUs。這可為具有 16 個 vCPUs 和 256 GB RAM 的六個 SQL Server Enterprise 伺服器提供足夠的容量。請參閱下列案例詳細資訊：

- 專用主機數量 – 2
- 執行個體系列 – x2iezn
- 每小時成本 (USD) – 11.009 美元
- 每月每單位成本 (USD) – 8,036 美元
- 實體核心 – 48

- 可用的 vCPU – 96
- 需要 Windows Server 核心授權 – 24
- SQL Server Enterprise 核心所需的授權 – 24
- 每月成本 – 16,073

兩個 X2iezn 系列 Amazon EC2 專用主機的總成本為每月 16,073 美元。如需定價的詳細資訊，請參閱此案例的 AWS 定價計算工具 [預估](#)。在這種情況下，組織可以利用其 Windows 授權，每月節省 1,755.65 USD。如果他們使用 Amazon EC2 專用主機，也可以減少所需的 SQL Server 授權數量。在共用租用中，他們需要 96 個 SQL Server Enterprise 授權，才能涵蓋六個 SQL Server Enterprise 伺服器，每個伺服器各有 16 vCPUs。不過，透過在實體核心層級使用 Amazon EC2 專用主機和授權，他們可以將所需的授權數量減少為 48 個核心。

下列詳細資訊會比較範例 3 的成本，並顯示相較於其他案例，使用 BYOL 選項在 Amazon EC2 專用主機上部署工作負載，可節省多少成本。

- 內部部署伺服器 – SQL Server
- vCPU – 16
- 記憶體 – 256
- 伺服器數量 – 6
- 案例 1 的每月成本：Windows (LI) + SQL Server Enterprise (LI) – 44,108 美元
- 案例 2 的每月成本：Windows (LI) + SQL Server Enterprise (BYOL) – 17,828 美元
- 案例 3 的每月成本：Amazon EC2 專用主機上的 Windows (LI) + SQL Server Enterprise (BYOL) – 16,073 美元

#### Note

成本以隨需定價為基礎。您可以使用 Savings Plans 或專用預留執行個體進一步降低成本。相較於隨需定價，這些選項提供彈性定價模式，可大幅節省成本。透過這些計劃，您可以承諾 1 或 3 年的期限。如需詳細資訊，請參閱本指南的 [Windows on Amazon EC2 最佳化支出](#) 一節。

請考慮下列 Amazon EC2 專用主機的付款選項：

- [專用主機](#) (Amazon EC2 文件)
- [專用主機預留](#) (Amazon EC2 文件)

- [Savings Plans](#) (Amazon EC2 文件 )

現在[AWS 定價計算工具](#)支援專用主機定價。這可協助您選擇適當的基礎專用主機。

## 成本最佳化建議

建議您採取下列後續步驟，使用最佳化成本 AWS Cost Explorer：

1. [啟用 Cost Explorer](#)。
2. 使用 Cost Explorer 來[檢視和分析 Amazon EC2 專用主機部署的成本和用量](#)。Amazon EC2
3. 驗證您正在執行 BYOL。您可以在 Amazon EC2 主控台的執行個體或 AMI 頁面上，或 describe-images 或 describe-instances 命令傳回的回應中，顯示下列平台詳細資訊和用量操作值。
  - 平台詳細資訊：Windows、用量操作：RunInstances：0002（包含授權）
  - 平台詳細資訊：Windows BYOL、用量操作：RunInstances：0800

## 其他資源

- [授權類型轉換的合格授權類型](#) (AWS License Manager 文件 )
- [AWS License Manager & 專用主機研討會](#) (AWS License Manager 研討會 )
- [Amazon EC2 專用主機FAQs](#) (AWS 文件 )
- [如何使用 VM Import/Export 從內部部署建立 Windows Server Bring-Your-Own-License AMIs](#) ( 部落格上的 AWS Microsoft 工作負載 )
- [VM Import/Export](#) (AWS 文件 )
- [Amazon Web Services 和 Microsoft：常見問答集](#) (AWS 文件 )
- [License Manager 中的授權類型轉換](#) (AWS License Manager 文件 )
- [在 Amazon EC2 專用主機上部署高可用性的 SQL Server](#) (AWS 雲端操作與遷移部落格 )

## 最佳化 Windows on Amazon EC2 的支出

### 概觀

將伺服器遷移至的其中一個首要考量 AWS 是基礎設施成本。事實上，雲端的好處之一是隨需支付資源，但有一些生產工作負載需要全年無休提供。[Savings Plans](#) 旨在節省 EC2 執行個體 AWS Lambda 和的穩定狀態 AWS 用量 AWS Fargate。

Savings Plans 提供靈活的定價模式，可協助您降低 Amazon EC2、Fargate、Lambda 和 Amazon SageMaker AI 用量的定價，以換取對一致用量（例如每小時 10 美元）的承諾。您承諾在一年或三年內維持一致的每小時運算花費量，並換取該用量的折扣。

您可以使用 Savings Plans 從三種不同的付款選項中進行選擇：

- 無預付款選項不需要任何預付款，而且您的承諾只會按月收費。
- 部分預付選項提供 Savings Plans 的較低價格。您至少需要預付一半的承諾，剩餘部分則每月收費。
- 所有預付選項提供最低價格，而且您的整個承諾只需支付一筆費用。

您可以追蹤 Savings Plans 過期和即將排入佇列的 Savings Plans AWS Cost Explorer。您可以使用 Savings Plans 提醒，在計劃過期日期前 1、7、30 或 60 天，或當承諾排入購買佇列時，接收預先電子郵件通知。這些通知也會在過期日期提醒您。您最多可以傳送通知給 10 個電子郵件收件人。

## 了解 Savings Plans

每種類型的運算用量都有隨需費率和 Savings Plans 費率。如果您承諾使用每小時 10 USD 的運算，則以 Savings Plans 費率取得所有用量的 Savings Plans 價格，最高可達 10 USD。運算支出承諾以外的任何用量都會按定期隨需費率計費。您可以在 [Cost Explorer](#) 中使用 [Cost Explorer](#) 來開始使用 Savings Plans AWS Management Console。

您可以使用 [Cost Explorer](#) 中提供的建議，輕鬆對 Savings Plans 做出承諾，以實現最大的節省。建議的每小時承諾取決於您的歷史隨需用量，以及您選擇的計劃類型、期限長度和付款選項。Savings Plans 會先套用至購買計劃的帳戶，然後與合併帳單系列中的其他帳戶共用。

### Note

預設 AWS Organizations 會啟用中的 Savings Plans 共用選項。您可以在付款人帳戶的 AWS Billing 主控台中拒絕此選項。您可以造訪[建議](#)頁面，查看 Savings Plans，其 AWS 建議協助您節省符合資格的用量。您可以隨時重新整理這些建議，以讓您輕鬆購買最佳的 Savings Plans。

## 計算節省計劃

Compute Savings Plans 提供最大的彈性，並協助您降低成本。無論執行個體系列、大小、可用區域、區域、作業系統或租用為何，這些計劃都會自動套用到 EC2 執行個體用量。它們也適用於 Fargate 或 Lambda 用量。例如，使用 Compute Savings Plans，您可以從 C4 變更為 M5 執行個體、將工作負載

從歐洲（愛爾蘭）轉移到歐洲（倫敦），或隨時將工作負載從 EC2 移至 Fargate 或 Lambda。您會自動繼續支付 Savings Plans 價格。

## EC2 Instance Savings Plans

EC2 Instance Savings Plans 提供最深層的折扣，以換取對區域中個別執行個體系列用量的承諾（例如，在維吉尼亞北部承諾一致的 M5 用量層級）。這會自動為您提供該區域中所選執行個體系列的隨需價格折扣，無論可用區域、大小、作業系統或租用。EC2 Instance Savings Plans 可讓您變更該區域中系列內執行個體之間的用量。例如，您可以從執行 Windows 的 c5.xlarge 移至執行 Linux 的 c5.2xlarge，並自動受益於 Savings Plans 價格。

Compute 和 EC2 Instance Savings Plans 都適用於屬於 Amazon EMR、Amazon Elastic Kubernetes Service (Amazon EKS) 和 Amazon Elastic Container Service (Amazon ECS) 叢集一部分的 EC2 執行個體。Savings Plans 不涵蓋 Amazon EMR、Amazon EKS 和 Amazon ECS 費用，但基礎 EC2 執行個體除外。EC2 Instance Savings Plans 會在 Compute Savings Plans 之前套用，因為 Compute Savings Plans 具有更廣泛的適用性。

### Note

做出承諾後，您無法輕鬆變更 Savings Plan。建議您在承諾使用 Savings Plans 任一選項之前，先仔細規劃。Savings Plans 提供比隨需定價更低的價格，以換取承諾，而且無法在該期間內取消。

## 每小時承諾範例

如果您購買 Savings Plan，您會對計劃期限做出每小時貨幣承諾。如果您承諾每小時 10 USD 的運算用量，Savings Plan 定價會自動套用至所有用量，每小時最多 10 USD。超出承諾的任何用量都會按一般隨需費率收費。您可以使用 Cost Explorer 中的 Savings Plans 購買建議工具，取得可最大限度地節省成本的建議承諾。特定計畫的每小時財務承諾無法在計畫期間內修改。如果您想要在分析用量後增加承諾，則可以購買額外的 Savings Plan 來涵蓋超額用量。

## Savings Plans 的優點

與預留執行個體相比，Savings Plans 提供更靈活的定價模式，可節省您的成本，同時充分利用 Savings Plans 提供的更廣泛運算選項。Savings Plans 提供折扣，即使您的運算需求變更也一樣。這可協助您跟上不斷變化的動態環境，而不會產生任何額外的管理開銷。以下是使用 Savings Plans 的一些其他好處：

- 易於使用 – 獲得自動折扣以換取貨幣承諾。

- 彈性 – 適用於多種用量類型的單一承諾。
- 潛在節省 – 有多種方式可以節省成本。請考量下列範例：
  - 使用 Compute Savings Plans ([d2.8xlarge、3 年、全部預付、視窗、共用租用、us-east-2](#)) 的 Windows Server 工作負載節省 60%
  - 使用 EC2 Instance Savings Plans 的 Windows Server 工作負載節省 73% ([d2.8xlarge、3 年、全部預付、視窗、共用租用、us-east-2](#))
  - 非極端執行個體類型 ([t3 系列、3 年、全部預付、視窗、共用租用、us-east-2](#)) 節省 28-41%
  - Windows Server 平均節省 25-40%

### Note

由於彈性降低，EC2 Instance Savings Plans 提供比 Compute Savings Plans 更高的折扣。您承諾以折扣價格使用。

每種類型的運算用量都有 Savings Plan 費率和隨需費率。下表顯示每種作業系統類型的 Savings Plans 和隨需費率。您需要為承諾用量支付 Savings Plans 費率，超過承諾的任何用量都會按一般隨需費率計費。

| 執行個體名稱        | Savings Plans 費率 | 隨需節省 | 隨需費率    | 作業系統                       | 區域            | 付款選項 | 期限長度 |
|---------------|------------------|------|---------|----------------------------|---------------|------|------|
| x2iedn.xlarge | 0.32 美元          | 61%  | 0.83 美元 | Linux                      | 美國東部 (維吉尼亞北部) | 不預付  | 3    |
| x2iedn.xlarge | 2.01 美元          | 50%  | 1.02 美元 | Windows                    | 美國東部 (維吉尼亞北部) | 不預付  | 3    |
| x2iedn.xlarge | 1.02 美元          | 20%  | 2.52 美元 | 包含 Windows 授權 + SQL Server | 美國東部 (維吉尼亞北部) | 不預付  | 3    |

| 執行個體名稱        | Savings Plans 費率 | 隨需節省 | 隨需費率    | 作業系統               | 區域            | 付款選項 | 期限長度 |
|---------------|------------------|------|---------|--------------------|---------------|------|------|
|               |                  |      |         | Enterprise Edition |               |      |      |
| x2iedn.xlarge | 0.32 美元          | 61%  | 0.83 美元 | BYOL               | 美國東部 (維吉尼亞北部) | 不預付  | 3    |

Savings Plans 包含作業系統，而且 BYOL 有個別的折扣。它們都在[Compute Savings Plans 計算器](#)中細分。

## 預留執行個體定價模型

AWS 根據稱為預留執行個體的承諾，有另一個定價模型。如果運算在您做出承諾之後變更，導致預留執行個體變成未使用，則此模型可能會發生問題。Savings Plans 旨在提供與[標準和可轉換預留執行個體](#)類似的成本降低，但具有更大的靈活性。Compute Savings Plans 提供 EC2 執行個體用量的較低價格，無論執行個體系列、大小、作業系統、租用或區域為何。它們也可以實現最大的靈活性。

下表可協助您選擇 Savings Plans 或預留執行個體。

|          | Reserved Instance | EC2 Instance Savings Plans | 計算節省計劃 |
|----------|-------------------|----------------------------|--------|
| 平均 1 年折扣 | 高達 38%            | 高達 29%                     | 高達 29% |
| 平均 3 年折扣 | 高達 58%            | 高達 73%                     | 高達 60% |
| 執行個體系列   | Fixed             | Fixed                      | 彈性     |
| 執行個體大小   | 已修正 (非 Linux)     | 彈性                         | 彈性     |
| 地理       | 1 區域              | 1 區域                       | 彈性     |
| 作業系統     | Fixed             | 彈性                         | 彈性     |

|        | Reserved Instance       | EC2 Instance Savings Plans | 計算節省計劃                    |
|--------|-------------------------|----------------------------|---------------------------|
| 服務     | Amazon EC2 或 Amazon RDS | Amazon EC2                 | Amazon EC2、Fargate、Lambda |
| 付款選項   | 全部、部分、無預付               | 全部、部分、無預付                  | 全部、部分、無預付                 |
| 執行個體限制 | 每個可用區域 20 個             | 沒有限制                       | 沒有限制                      |

### Note

Savings Plans 的運作方式是根據每小時貨幣承諾為您提供折扣。您無法在計劃期限內取消或變更每小時財務承諾，但您可以購買額外的 Savings Plans 來涵蓋額外的用量。這可讓您在機群成長時保持一致的每小時承諾。

您可以使用 [AWS Cost Explorer](#) 或 [AWS 雲端 Intelligence Dashboards](#) 等工具來追蹤您的承諾。Cost Explorer 提供涵蓋範圍目標行，可協助您的組織規劃其 Savings Plans 涵蓋範圍策略。如果 75% 的工作負載處於穩定狀態，則 75% 是良好的目標。根據動態工作負載，這會留下 25% 的隨需/可變支出。如果您需要將提高到 85% 的涵蓋範圍，您可以購買另一個 Savings Plans 承諾來提高每小時貨幣承諾。

### Note

我們建議您購買 Savings Plans 而非預留執行個體，但如果您已購買預留執行個體，這兩個承諾模型可以一起運作。

請考慮您購買預留執行個體，但您想要開始嘗試 Savings Plans 選項的範例。此組合有邏輯可套用至您的最終帳單。以下是您可以套用至的階層 AWS 帳戶：

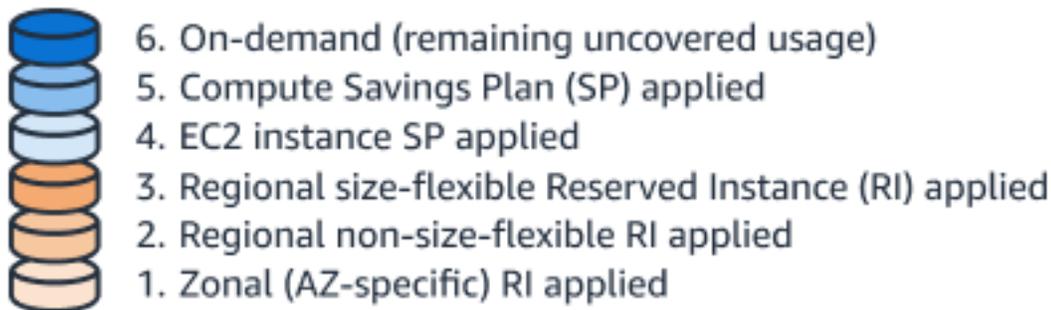
1. 區域預留執行個體適用於擁有該執行個體的帳戶。如果預留執行個體還剩小時，則適用於組織的其餘部分。
2. 適用於 Windows 的非彈性大小區域預留執行個體適用於擁有該執行個體的帳戶上的相符用量。任何剩餘的項目都會推展到組織的其餘部分。

3. 彈性大小的區域預留執行個體會套用至擁有該執行個體的帳戶（系列中最小的執行個體會先進入較大的執行個體），然後再套用至組織的其餘部分。
4. 區域預留執行個體適用於任何未使用的隨需容量保留。
5. EC2 Instance Savings Plans 適用於購買它的帳戶。
6. Compute Savings Plans 適用於購買它的帳戶。

### Note

折扣從產生最高折扣的用量開始，然後降至最低折扣。對於最常見的執行個體類型（例如 T3, M6 和 C5），傳統上 Windows 執行個體的折扣潛力低於 Linux。這表示在大多數情況下，Linux 執行個體的效益超過 Windows 執行個體。

下圖顯示將預留執行個體與 Savings Plans 分割後的價格。運算和 EC2 Instance Savings Plans 都適用於執行中的執行個體，然後適用於未使用的隨需容量保留。



## 成本最佳化案例

本節涵蓋使用包含授權計費模型的 Amazon EC2 專用主機和 Amazon EC2 執行個體的成本最佳化案例。

### Amazon EC2 專用執行個體

考慮您將遷移現場部署 Windows 工作負載的目標案例 AWS。您的資料中心有下列伺服器：

- 兩個伺服器，具有 16 個 vCPU 和 128 GB RAM
- 兩個伺服器，具有 32 vCPU 和 164 GB RAM
- 一個具有 8 個 vCPU 和 64 GB RAM 的伺服器
- 16 部伺服器搭配 vCPU 和 32 GB RAM

此外，假設您可以將自己的授權帶到，AWS 因為您有足夠的授權可接管。下表顯示您可以在其中使用的伺服器執行個體 AWS。

| 執行個體類型     | CPU | RAM | Amount (數量) |
|------------|-----|-----|-------------|
| r5.4xlarge | 16  | 128 | 2           |
| r5.8xlarge | 32  | 256 | 2           |
| r5.2xlarge | 8   | 64  | 1           |
| r5.xlarge  | 4   | 32  | 16          |
|            |     |     | 21          |

分析顯示這 21 個虛擬機器可以使用 R5 執行個體系列主機分散到兩個專用主機。下表顯示這兩個專用主機的コスト。

| 專用主機隨需案例              | 預付款       | 1 個月      | 1 年        | 3 年        | AWS 定價計算工具                    |
|-----------------------|-----------|-----------|------------|------------|-------------------------------|
| 隨需                    | 無         | 10,123 美元 | 121,475 美元 | 364,392 美元 | <a href="#">AWS 定價計算工具 預估</a> |
| 1 年 Savings Plan      | 無         | 7,447 美元  | 89,362 美元  | –          | <a href="#">AWS 定價計算工具 預估</a> |
| 3 年 Savings Plan      | 無         | 5,476 美元  | 65,712 美元  | 197,128 美元 | <a href="#">AWS 定價計算工具 預估</a> |
| 3 年 Savings Plan 含預付款 | 84,438 美元 | 2,755 美元  | 117,499 美元 | 183,618 美元 | <a href="#">AWS 定價計算工具 估計</a> |

如果您有想要遷移的伺服器 AWS，一年 Savings Plan 的最終價格為 89,362 美元，而不是隨需價格 121,475 美元。這代表一年後有 26.5% 的折扣。如果您考慮長期留在 AWS，則可以選擇 3 年 Savings Plan，以節省更深層的成本。在三年結束時，您需支付 197,128 USD，而不是 364,392 USD。這可節省三年後 46% 的總金額。

## 包含授權的 Amazon EC2 執行個體

假設您要將單一三層應用程式遷移至其中 AWS，而且您想要使用提供的授權 AWS。此外，假設您的應用程式適用於下列伺服器：

- 具有兩個 vCPUs 兩個 Web 伺服器
- 兩個應用程式伺服器，具有八個 vCPUs 和 16 GB RAM
- 具有 16 個 vCPUs 和 64 GB RAM 的兩個資料庫伺服器（使用 SQL Server Standard Edition）

下表顯示您可以在中使用的伺服器執行個體 AWS。

| 執行個體類型     | CPU | RAM | Amount (數量) |
|------------|-----|-----|-------------|
| c5.large   | 2   | 4   | 2           |
| c5.2xlarge | 8   | 16  | 2           |
| r5.2xlarge | 8   | 64  | 2           |
|            |     |     | 6 個伺服器      |

下表顯示這些伺服器的成本 AWS。

| 包含的授權 AWS               | 預付款 | 1 個月     | 1 年       | 3 年        | AWS 定價計算工具                    |
|-------------------------|-----|----------|-----------|------------|-------------------------------|
| 隨需                      | 無   | 3,912 美元 | 46,950 美元 | 140,849 美元 | <a href="#">AWS 定價計算工具 預估</a> |
| 1 年 Savings Plan        | 無   | 3,466 美元 | 41,952 美元 |            | <a href="#">AWS 定價計算工具 預估</a> |
| 3 年 Savings Plan, 無需預付款 | 無   | 3,189 美元 | 38,264 美元 | 114,804 美元 | <a href="#">AWS 定價計算工具 預估</a> |

| 包含的授權<br>AWS                 | 預付款            | 1 個月 | 1 年 | 3 年 | AWS 定價計<br>算工具                     |
|------------------------------|----------------|------|-----|-----|------------------------------------|
| 3 年 Savings<br>Plan 含預付<br>款 | 112,110 美<br>元 | 無    | 無   | 無   | <a href="#">AWS 定價計<br/>算工具 估計</a> |

如果您想要針對隨需定價的生產環境（全年無休）執行這些伺服器，您每月需支付 3,912 USD 的費用。支付此每月成本相當於一年後 46,950 美元，三年後總計 140,849 美元。

如果您選擇 1 年期 Savings Plan 而沒有預付款，每月成本會降低到 3,466 美元。在第一年結束時，您需支付 41,952 USD。這是 11% 的總折扣。如果您選擇 3 年期 Savings Plan 而沒有預付款，每月成本會降低到 3,189 美元。在三年結束時，您需支付 114,804 USD。這可讓您節省 18.5% 的費用。

## 成本最佳化建議

這兩種情況都可協助您在 中規劃和預測工作負載時節省成本 AWS。請務必了解，第二個案例中的折扣比第一個案例少。在第二個案例中，授權價格包含在雲端伺服器的價格中。AWS 不會提供授權價格的折扣，但您可以始終使用授權（在特定情況下），並 AWS 始終保證最佳的運算/執行個體價格。

建議您執行下列動作，以控制運算和執行個體資源上的 AWS 花費：

- 存取建議
- 根據您的需求自訂建議
- 檢閱每小時承諾

### 存取建議

您可以使用 [Amazon EC2 主控台](#) 來存取 Savings Plan 的建議。您甚至可以下載建議，以供日後檢閱 CSV 格式。如需詳細資訊，請參閱 [Savings Plans 文件中的監控 Savings Plans](#)。

### 根據您的需求自訂建議

開啟 [Amazon EC2 主控台](#)，展開執行個體區段，然後選擇 Savings Plans。此頁面會顯示建議前後的執行個體和運算定價。您也可以針對建議調整下列因素：

- 期限 – 例如，1–3 年

- 付款選項 – 例如預付、部分預付或無預付
- 歷史記錄 – 例如，過去 7、30 或 60 天

## 檢閱每小時承諾

使用相同的範例，假設您的執行個體全年無休執行。建議是使用 Savings Plan。根據大小，您的隨需價格為每小時 120 美元。您可以選擇每小時遞交 90 USD，但這可能會根據區域、執行個體和購買選項而有所不同。在此範例中，相較於隨需成本，您可以節省 25%。您也可以追蹤您的使用率和涵蓋範圍，如果低於您定義的閾值，並在預算即將結束時設定提醒。

## 檢閱建議

我們建議您仔細檢閱 Savings Plan 建議。AWS 不會在未經您許可的情況下變更任何內容。這些只是建議，由您決定是否套用。

## 購買計劃

開啟 [Amazon EC2 主控台](#)，展開執行個體區段，然後選擇 Savings Plans。然後，選擇購買 Savings Plans。根據您的需求，您可以選取下列選項：期限、區域、執行個體系列、每小時承諾、付款選項，甚至是開始日期。您可以選擇 Compute Savings Plans、EC2 Instance Savings Plans 和 SageMaker AI Savings Plans。如需詳細資訊，請參閱 [Savings Plans 文件中的購買 Savings Plans](#)。

## 取得使用率報告

購買 Savings Plan 之後，您可以取得使用率報告。報告可協助您檢查使用率、查看購買的計劃是否足以涵蓋折扣並最大化折扣，以及取消或新增折扣。此報告可以匯出為其他格式，例如 CSV。如需詳細資訊，請參閱 Savings Plans 文件中的 [使用使用率報告](#)。

## 遵循購買最佳實務

建議您在購買 Savings Plans 之前遵循這些最佳實務：

- 使用 [AWS Trusted Advisor](#) 移除閒置的 EC2 資源。
- 在購買 Savings Plans 之前執行任何適當的規模調整。
- 建立您持續保留 30-60 天的每小時費率。
- 購買承諾，以涵蓋組織熟悉的一致每小時費率。考慮需求或季節的波動。
- 選擇每季檢閱 Savings Plans 預算以維持一致的費率（例如，Savings Plans 涵蓋範圍的 70% 涵蓋範圍目標）。如果費率低於所需的涵蓋範圍，請購買額外的 Savings Plan 做為校正，以符合您的涵蓋範圍目標。

## 其他資源

- [Amazon EC2 預留執行個體的 Savings Plans](#) (AWS 白皮書 )
- [了解 Savings Plans 如何套用至您的 AWS 用量](#) (Savings Plans 文件 )
- [宣布 EC2 Windows Server 和 SQL Server 執行個體的每秒帳單](#) (AWS 文件 )
- [AWS 成本最佳化系列：Savings Plans 影片 | Amazon Web Services](#) (YouTube)

## 使用 AWS 工具監控成本

### 概觀

成本可見性是最佳化成本的關鍵因素 AWS。AWS 有許多工具可用來視覺化成本，並建立提醒以回應這些成本。其中包括可協助您追蹤和報告支出的工具，例如 AWS Budgets。本節涵蓋監控 Windows AWS 支出的特定方式，因此您可以追蹤並據此回應預算需求。這包括將必要的標籤新增至您的 Windows EC2 資源。這些標籤可讓您使用 正確監控 Windows EC2 和其他 Microsoft 服務 AWS Budgets。

透過監控支出和使用 AWS 工具建立提醒，您可以更了解目前的支出、預計支出和異常支出。如果您使用 [Savings Plans](#) 協助降低每小時 EC2 執行個體定價，建議您檢視 Savings Plan 的整體使用率和涵蓋範圍。這可協助您確保持續實現節省。您可以使用 AWS Cost Explorer 來檢視 Savings Plan 庫存，並根據先前的用量取得其他 Savings Plans 的建議。您也可以使用 [AWS Budgets](#) 和設定 來追蹤特定花費 [AWS Cost Anomaly Detection](#)。

### 成本最佳化建議

建議您採取下列後續步驟，使用 AWS Budgets、Cost Explorer 和異常偵測來最佳化成本：

- 標記 Windows EC2 資源
- 使用 設定提醒 AWS Budgets
- 啟用成本異常偵測
- 取得即時支出分析
- 使用 Cost Explorer 檢視 Windows 包含授權的支出

## 標記 Windows EC2 資源

若要有效監控 AWS 支出，您必須為要監控的工作負載建立[標記策略](#)。這很重要，以便您可以分類分組資源，並收到特定支出的通知，而不是一般用量支出。您可以使用標記資源，這些資源不僅有助於降低成本，也可以用於其他用途，例如[AWS Systems Manager 自動化](#)。此外，我們建議您為[必要的標籤](#)實作一些管理。

若要追蹤您在 AWS Budgets Cost Explorer 和 Cost Anomaly Detection 中的花費，您必須確保有適當的標籤。您可以使用標籤為符合這些標籤的項目設定特定預算，以便在花費增加時收到提醒。

例如，您可以使用簡單的標籤，例如 Key=OS Value=Windows。這會將所有 Windows 執行個體放在一個群組中，您可以追蹤其花費。您也可以將標籤用於其他項目，例如 Systems Manager。建立標籤後，您必須啟用標籤以進行成本追蹤。請考慮新增 [AWS Config 規則，以監控連接至特定資源的標籤](#)。如果執行中的資源不包含適當的標籤，AWS Config 可以提醒您，這可讓您準確呈現 Windows EC2 花費。

標籤就位後，您可以在 [中](#) 建立自訂預算 AWS Billing。這可讓您了解 Windows EC2 花費。您可以設定每日預算或每月預算。

### 使用 設定提醒 AWS Budgets

在此範例案例中，您會為 Windows EC2 建立每日預算。這是一個經常性預算，使用自動調整選項來追蹤您的支出並相應地調整預算。如果您有靜態環境，您可以改用固定預算。請務必選擇基準時間範圍（例如 30 天）。

1. 登入 AWS Management Console 並開啟 [AWS Cost Management 主控台](#)。
2. 在導覽窗格中，選擇 Budgets (預算)。
3. 在頁面頂端，選擇 Create budget (建立預算)。
4. 在預算設定下，選擇自訂 (進階)。
5. 在預算類型下，選擇成本預算。然後選擇下一步。
6. Under Details、forBudget 名稱，輸入預算的名稱。例如，Windows EC2 支出。
7. 在設定預算金額下，針對期間，選擇每日。
8. ForBudget 續約類型，針對預算期間之後重設的預算選擇經常性預算。
9. 針對開始日期，選擇開始日期或期間，開始追蹤預算金額。
10. ForBudgeting 方法，選擇自動調整 (新增)。
11. 針對基準時間範圍，選擇自訂範圍，然後輸入 30 天。
12. 選擇 Next (下一步)。

13. 在預算範圍區段中，選取篩選特定 AWS 成本維度。這是標籤用來建立適當維度的地方。AWS Budgets 不支援平台類型作為其篩選條件中的選項。因此，您必須套用作業系統標籤。
14. 選擇新增篩選條件，然後從維度中選取標籤選項。
15. 選擇作業系統標籤，然後選擇此標籤的 Windows 值，以建立標籤的預算。
16. 選擇 Next (下一步)。
17. 在設定提醒頁面上，選擇新增提醒閾值。在這裡，您設定兩個提醒：一個用於 50% 閾值，另一個用於 100% 閾值。如果超過 50% 閾值警示的時間超過當月中點，則會提供警告。如此一來，您就可以檢查支出是否超出預期，並在月底之前做出反應。
18. 針對閾值，輸入 50 並選取預算金額的 %。
19. 針對觸發，選擇實際。
20. 針對電子郵件收件人，輸入電子郵件地址。為閾值 100 新增另一個提醒。

#### Note

此範例使用提醒的電子郵件通知，但您也可以使用其他方法，例如 [Slack](#)。

## 啟用成本異常偵測

您可以使用成本標籤來設定異常的支出提醒。例如，您可以使用 [AWS Cost Anomaly Detection](#) 為您的花費建立監控，並在系統偵測到帳戶中的異常花費時收到提醒。

若要為先前建立的 Key=OS 和 Value=Windows 標籤設定監視器和警示，請執行下列動作：

1. 登入 AWS Management Console 並開啟 [AWS Cost Management 主控台](#)。
2. 在導覽窗格中，選擇 Cost Anomaly Detection (成本異常偵測)。
3. 選擇成本監控索引標籤，然後選擇建立監控。
4. 在步驟 1 中，選擇成本分配標籤作為您的監視器類型。
5. 針對成本分配標籤金鑰，選擇 Windows EC2 花費。
6. 針對成本分配標籤值，選擇 Windows。
7. 在命名監視器中，輸入 Windows EC2 支出。
8. 選擇 Next (下一步)。
9. 若要為提醒建立訂閱，請選取建立新訂閱。如果您有現有的訂閱，請選取 Choose an existing subscription (選擇現有的訂閱)。
10. 針對訂閱名稱，輸入 Windows EC2 支出異常。

- 11 針對提醒頻率，選擇每日摘要。
- 12 對於提醒收件人，請輸入您的電子郵件地址。
- 13 選擇新增閾值。針對閾值，輸入 10，然後選取高於預期速度的百分比。
- 14 選擇 Create monitor (建立監視器)。

## 取得支出的即時檢視

提醒是監控 Windows EC2 花費的實用工具，但如果您想要即時檢視花費，則必須使用 Cost Explorer。觀看此影片以了解 Cost Explorer 如何讓您分析和降低 EC2 成本。如需詳細資訊，請觀看 YouTube 上的 [AWS 支援 | 了解並降低 EC2 成本](#) 影片。

## 檢視 Windows 的授權包含支出

您可以使用 Cost Explorer 來檢視帳戶中的 EC2 Windows 花費。若要查看 Windows 的授權包含支出，您必須在 Cost Explorer 中設定下列正確的 [篩選條件](#)：

- 針對平台，選擇 Windows (Amazon VPC)。針對 API 操作，選擇 RunInstance : 0002。這是包含授權之 Windows EC2 執行個體的 AWS Billing 程式碼。
- 如果您想要檢視 BYOL 執行個體花費，請將 RunInstance : 0002 變更為 RunInstance : 0800。這是 Windows EC2 BYOL 的帳單代碼。

透過 Cost Explorer 中的這種可見性，您可以快速篩選成本，確切地篩選您在 Windows EC2 上的支出。如果您想要深入了解 AWS 支出，您可以使用 AWS Cost and Usage Report 篩選個別執行個體層級的支出。您也可以產生可在 Amazon QuickSight 中視覺化的報告，並建置自訂儀表板。

如需詳細資訊，請觀看 YouTube 上的 [AWS 支援 - 視覺化您的成本和用量報告影片](#)。

## 其他資源

- [使用 設定必要的標籤 AWS Config](#) (AWS Config 文件 )
- [AWS Budgets 教學課程 - 設定提醒 AWS Billing | Amazon Web Services](#) (YouTube)
- [AWS Cost and Usage Report 查詢程式庫](#) (AWS Well-Architected 實驗室 )

# SQL Server

客戶在上執行 Microsoft AWS 工作負載的時間已超過 15 年，比任何其他雲端供應商都長。這主要是因為 AWS 在雲端中擁有最多的 Microsoft 應用程式體驗，並在下列領域提供最適合 Windows Server 和 Microsoft SQL Server 的平台：

- 更高的效能和可靠性
- 更高的安全性和身分服務
- 更多遷移支援
- 最廣泛且最深入的功能
- 降低總擁有成本 (TCO)
- 彈性的授權選項

AWS 支援建置和執行依賴 SQL Server 的 Windows 應用程式所需的一切，包括 Active Directory、.NET、SQL Server、Windows 桌面即服務，以及所有支援的 Windows Server 版本。透過經過驗證的專業知識，AWS 可協助您輕鬆提升和轉移、重構甚至現代化 Windows 工作負載。

本指南的本節涵蓋下列主題：

- [選擇高可用性和災難復原解決方案](#)
- [了解 SQL Server 授權](#)
- [為 SQL Server 工作負載選取正確的 EC2 執行個體](#)
- [合併執行個體](#)
- [比較 SQL Server 版本](#)
- [評估 SQL Server 開發人員版本](#)
- [評估 Linux 上的 SQL Server](#)
- [最佳化 SQL Server 備份策略](#)
- [現代化 SQL Server 資料庫](#)
- [最佳化 SQL Server 的儲存](#)
- [使用 Compute Optimizer 最佳化 SQL Server 授權](#)
- [使用 Compute Optimizer 最佳化 SQL Server 大小](#)
- [檢閱 SQL Server 工作負載 Trusted Advisor 的建議](#)

# 選擇高可用性和災難復原解決方案

## 概觀

建議您為上的 SQL Server 部署設計架構 AWS，以滿足業務需求，同時滿足[災難復原 \(DR\) 目標](#)，包括復原時間目標 (RTO) 和復原點目標 (RPO)。下列解決方案可協助您為 Amazon Elastic Compute Cloud (Amazon EC2) 上的 SQL Server 設計正確的架構，同時最佳化 SQL Server 工作負載的成本。

- SQL Server Always On 可用性群組 – SQL Server Always On 可用性群組可為 SQL Server 資料庫提供高可用性和災難復原 (HA/DR) 解決方案。可用性群組由一組一起容錯移轉的使用者資料庫組成。Always On 可用性群組也在資料庫層級提供備援，但不需要共用儲存，每個複本都有自己的本機儲存。您可以將此功能部署為 HA/DR 解決方案。如需詳細資訊，請參閱 Microsoft 文件中的[什麼是 Always On 可用性群組？](#)。
- SQL Server Always On 容錯移轉叢集執行個體 (FCI) – SQL Server Always On FCIs 使用 Windows Server 容錯移轉叢集 (WSFC) 在 SQL Server 執行個體層級提供 HA。FCIs 需要共用儲存來託管資料庫。您可以使用共用區塊儲存或共用檔案儲存。例如，您可以使用 Amazon FSx for Windows File Server 或 Amazon FSx for NetApp ONTAP 做為具有多個可用區域的共用儲存解決方案。如需詳細資訊，請參閱 Microsoft 文件中的[Always On 容錯移轉叢集執行個體 \(SQL Server\)](#)。
- SIOS DataKeeper – SIOS DataKeeper 可透過啟用跨越可用區域和的 SQL Server FCI，協助您同時滿足 HA 和 DR 需求 AWS 區域。SIOS DataKeeper 會使用本機 Amazon Elastic Block Store (Amazon EBS) 磁碟區建立叢集虛擬 SAN，並使用 HA 可用區域之間的不同步複寫，同時使用區域和之間的非同步複寫進行災難復原。如需詳細資訊，請參閱 SIOS 文件中的[Windows 應用程式的高可用性保護](#)。
- 分散式可用性群組 – 分散式可用性群組是一種特殊類型的可用性群組，跨越兩個不同的 Always On 可用性群組。可用性群組可以位於兩個不同的區域（例如 us-east-1 和 us-west-1）。您可以將分散式可用性群組視為可用性群組的可用性群組，因為基礎 Always On 可用性群組是在兩個不同的 WSFC 叢集上設定。部署分散式可用性群組需要 SQL Server Enterprise Edition。如需詳細資訊，請參閱 Microsoft 文件中的[分散式可用性群組](#)。
- 日誌運送 – 您可以實作日誌運送來保護跨多個區域的資料庫，在極少數情況下，區域會受到影響且無法使用。根據交易和日誌運送頻率，您可以在幾分鐘內達到 RPO 和 RTO。如需詳細資訊，請參閱 Microsoft 文件中的[關於日誌運送 \(SQL Server\)](#)。
- AWS Elastic Disaster Recovery – Elastic Disaster Recovery 是一種軟體即服務 (SaaS) 應用程式，可針對 DR 目的管理從任何基礎設施到 AWS 的伺服器複寫。您也可以使用 Elastic Disaster Recovery 跨區域複寫 SQL Server。Elastic Disaster Recovery 是一種代理程式型解決方案，可將包括作業系統、所有安裝的應用程式和所有資料庫在內的整個虛擬機器複寫到預備區域。如需詳細資訊，請參閱[Elastic Disaster Recovery 文件中的什麼是 Elastic Disaster Recovery？](#)。

- AWS Database Migration Service (AWS DMS) – AWS DMS 支援資料往返即時遷移 AWS，包括不同的區域。您可以使用此功能，在不同區域中設定個別的 SQL Server 執行個體，做為災難復原資料庫。如需詳細資訊，請參閱 AWS DMS 文件中的[什麼是 AWS Database Migration Service ?](#)。

## SQL Server Always On 可用性群組

如果您只將 SQL Server Enterprise Edition 用於高可用性 [Always On 可用性群組](#)，則可以利用基本可用性群組降級為 SQL Server Standard Edition。您可以使用基本可用性群組，而不是 Always On 可用性群組，將成本從 65% 降低到 75%。

### Note

如需不同 SQL Server 版本之間成本差異的其他資訊，請參閱本指南的[比較 SQL Server 版本](#)一節。

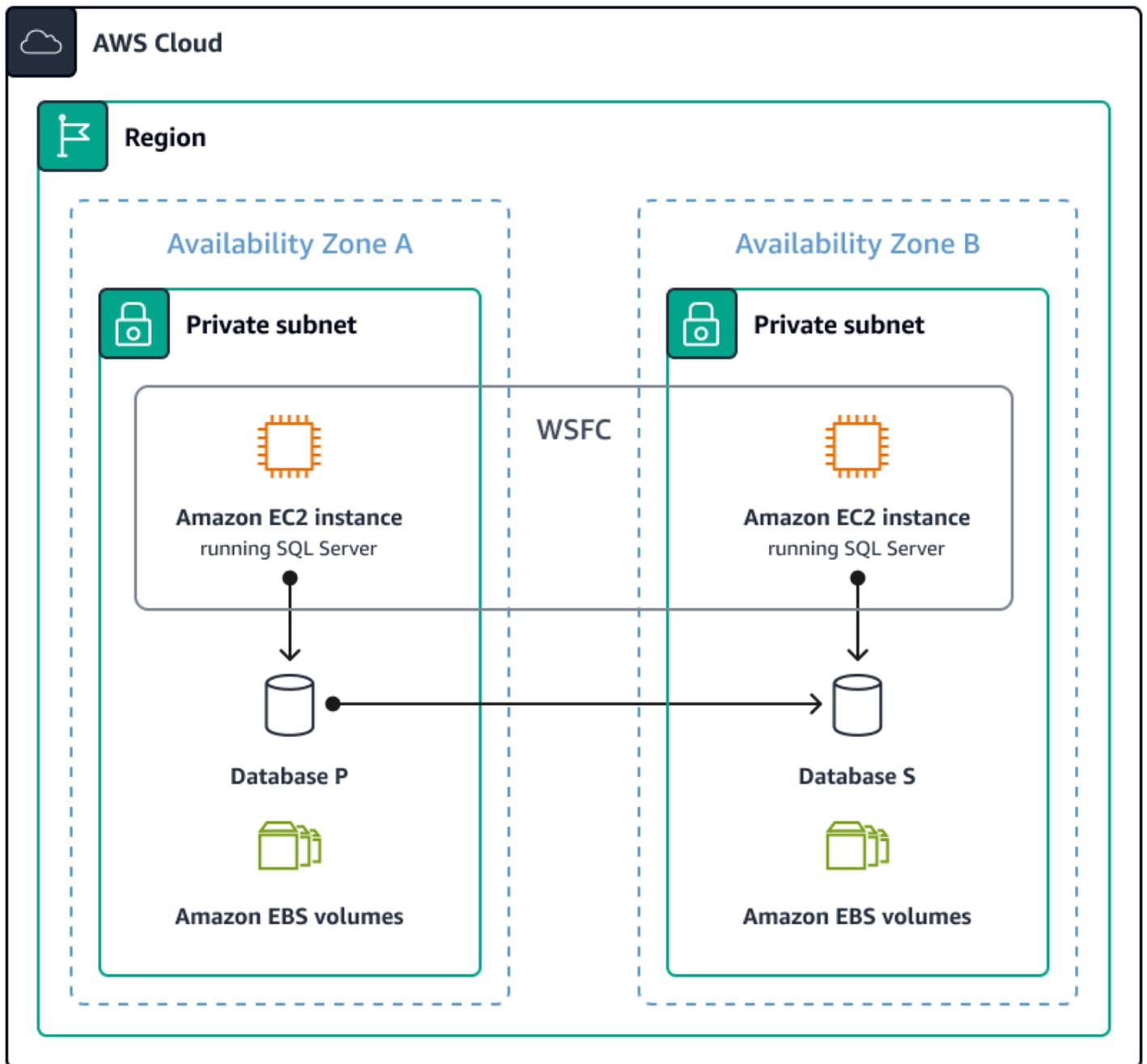
### 功能

- SQL Server Standard Edition 中提供
- 兩個複本的限制（主要和次要）
- 次要複本上沒有讀取存取權
- 次要複本上沒有完整性檢查

### 限制

- 每個可用性群組僅支援一個可用性資料庫
- 基本可用性群組不能是分散式可用性群組的一部分

下圖顯示 Windows Server 容錯移轉叢集解決方案的範例架構。



## SQL Server Always On 容錯移轉叢集執行個體

您可以使用容錯移轉叢集執行個體 (FCIs) 來確保持續的資料庫操作，同時將停機時間降至最低，並降低資料遺失的風險。如果您正在尋找 SQL Server 資料庫的高可用性，而沒有僅供讀取複本組態，FCIs 會提供可靠的解決方案。

與可用性群組不同，FCIs 可以提供可靠的容錯移轉解決方案，而不需要 SQL Server Enterprise Edition。相反地，FCIs 只需要 SQL Server Standard Edition 授權。您可以使用 FCIs 將 SQL Server 授權成本降低 65-75%。

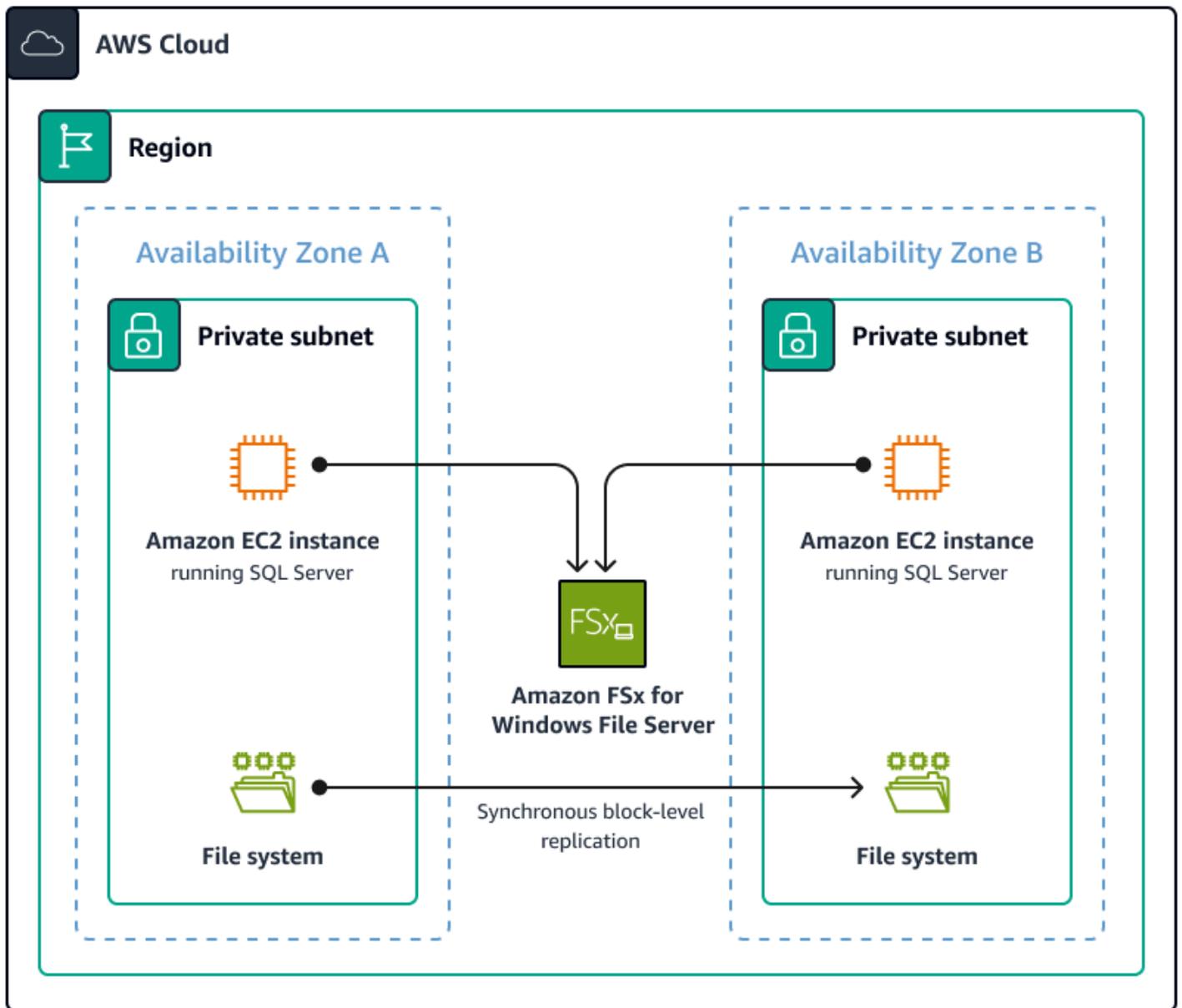
**Note**

如需 SQL Server 版本之間成本差異的其他資訊，請參閱本指南的[比較 SQL Server 版本](#)一節。

考慮下列各項：

- Amazon FSx for Windows File Server 提供功能強大的解決方案，可滿足您的 SQL Server FCI 共用儲存需求。您可以使用 FSx for Windows File Server 來避免購買儲存複寫解決方案的授權，並自行管理共用儲存。這可以大幅節省 30-40% 的成本。如需詳細資訊，請參閱 AWS Storage Blog 上的[使用 Amazon FSx for Windows File Server 簡化 Microsoft SQL Server 高可用性部署](#)文章。
- 透過[軟體保證利益摘要](#)（可下載的 PDF）和自攜授權 (BYOL) 模型，只要次要伺服器是被動伺服器，您就可以利用被動容錯移轉利益。這可節省 SQL 授權的成本，因為您不必向叢集的被動節點提供授權。

下圖顯示使用 FSx for Windows File Server 的 SQL Server FCI 架構範例。



## SIOS DataKeeper

如果您打算部署 SQL Server FCIs，建議您考慮共用儲存需求 AWS。傳統的內部部署安裝通常會使用儲存區域網路 (SAN) 來滿足共用儲存需求，但這不是可行的選項 AWS。Amazon FSx for Windows File Server 是建議的 SQL Server FCI 儲存解決方案 AWS，但其限制會防止在不同中新增叢集伺服器 AWS 區域。

您可以使用 [SIOS DataKeeper](#) 來建立同時涵蓋可用區域和區域的 SQL Server FCI，同時將成本降低 58-71%。SIOS DataKeeper 可協助您實現 FCI 的高可用性優勢。這使得 SIOS DataKeeper 成為組織經濟實惠且可靠的解決方案。

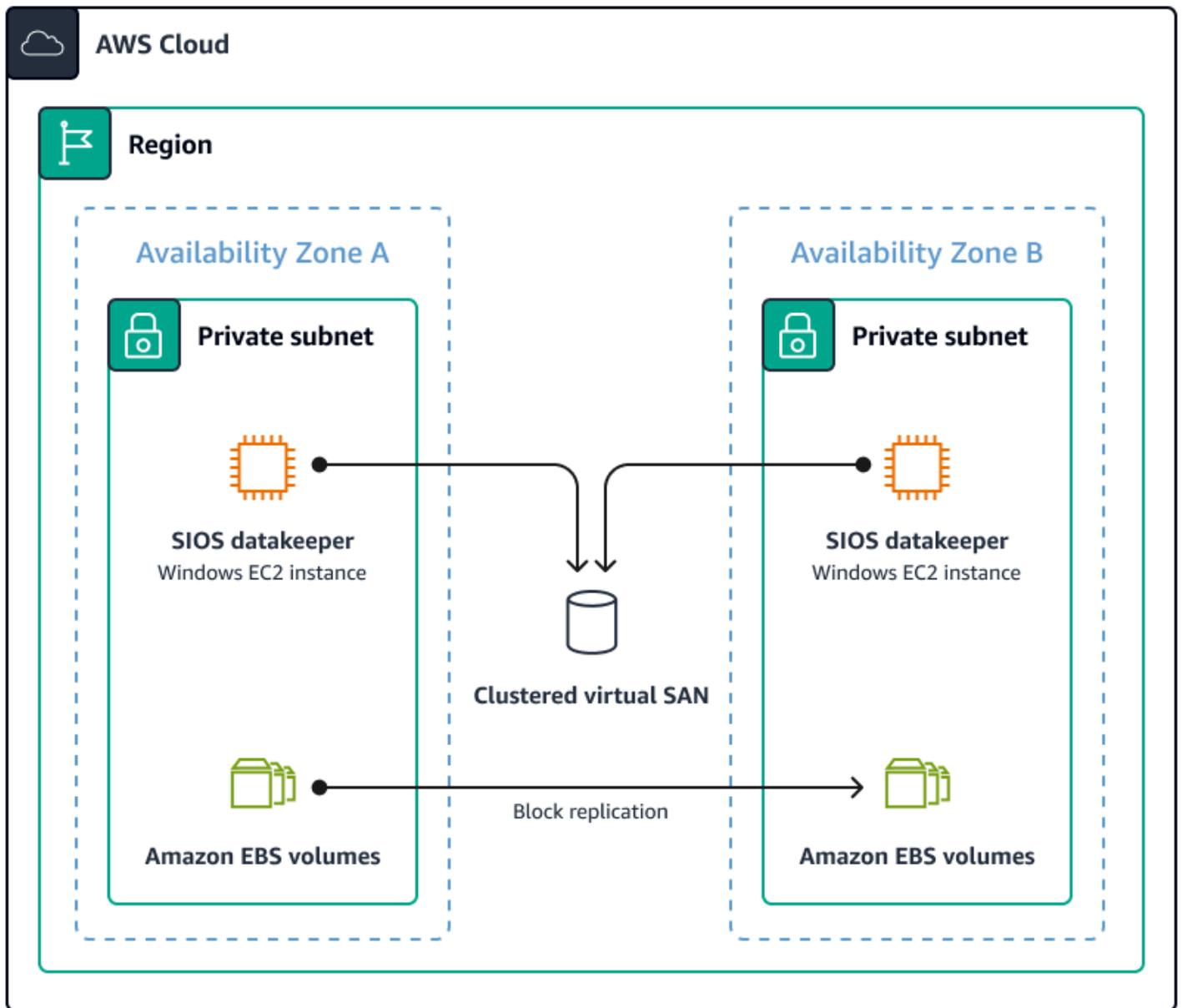
請考慮使用 SIOS DataKeeper 的下列其他優點：

- SIOS DataKeeper 會使用本機 EBS 磁碟區建立叢集虛擬 SAN，並在可用區域之間使用同步複寫以獲得高可用性。為了進行災難復原，SIOS DataKeeper 會使用區域之間的非同步複寫。
- SIOS DataKeeper 使用 SQL Server Standard Edition 提供企業級叢集功能。相較於使用 SQL Server Enterprise Edition 的 SQL Server Always On 可用性群組實作高可用性，這可將 SQL Server 授權成本降低 65-75%。使用 SIOS DataKeeper，您可以建立高度可用、靈活且符合成本效益的 SQL Server 環境，以滿足組織的需求。

**Note**

如需 SQL Server 版本之間成本差異的其他資訊，請參閱本指南的[比較 SQL Server 版本](#)一節。

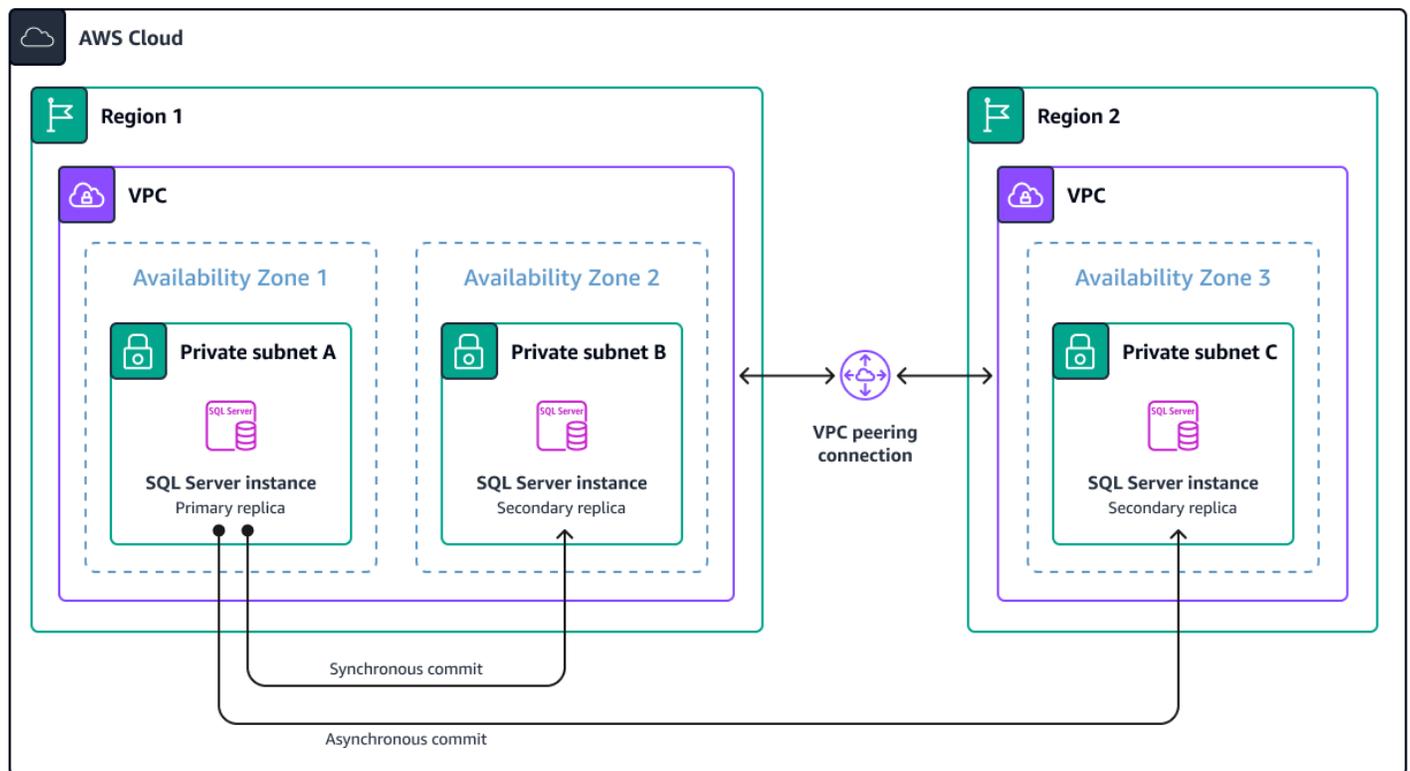
下圖顯示使用叢集虛擬 SAN 解決方案的 SQL Server FCI 範例架構。



## Always On 可用性群組

您可以將 Always On 可用性群組用於高可用性和災難復原目的。您可以透過在一個區域中的兩個可用區域部署 SQL Server 來實現高可用性。您可以透過跨區域擴展可用性群組來實現災難復原。

下圖顯示以 Always On 可用性群組為基礎的解決方案架構範例。圖表區域 1 中的複本使用同步遞交，可提供可用性群組的自動容錯移轉。區域 2 中的複本正在使用非同步遞交，這將需要可用性群組的手動容錯移轉。

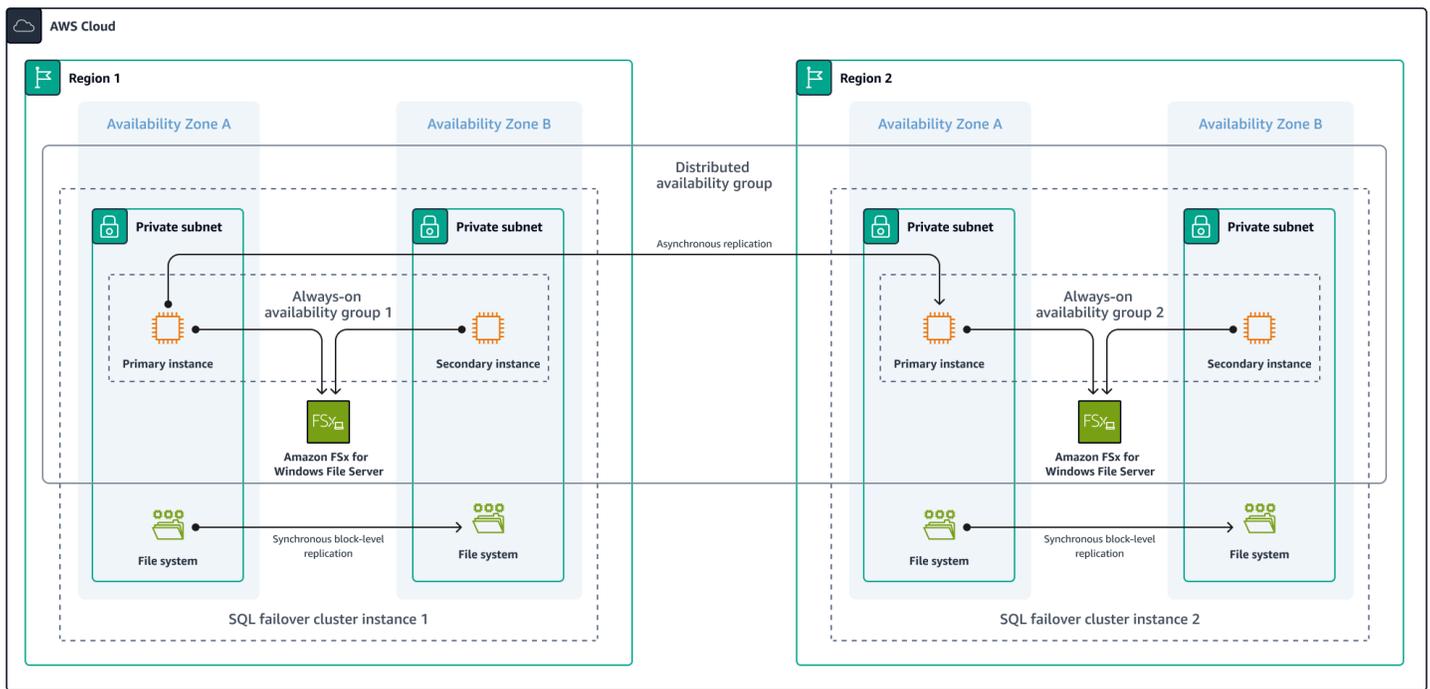


## 分散式可用性群組

對於在可靠性或災難復原方面無法犧牲的任務關鍵 SQL Server 部署，我們建議採用多區域方法。將您的可用性群組分散到多個區域是最有彈性的解決方案，可維持業務連續性和將停機時間降至最低。

此架構充分利用 Amazon FSx for Windows File Server 的功能，包括共用儲存、同步區塊層級複寫和 SQL Server FCIs。這些功能可讓您建立跨多個可用區域的高可用性 SQL Server 環境。透過在另一個區域中複寫此設定，您可以取得完全備援的系統，即使是最嚴重的中斷也能處理。此解決方案的區別在於其提供的彈性和安全性層級。分散式可用性群組的網域獨立架構可讓基礎 Windows 叢集伺服器加入不同的 Active Directory 網域，而憑證型身分驗證可確保 SQL Server 環境的最大保護，並為多區域 DR 策略提供高 RTO 和 RPO 需求。如需有關建置多區域架構的資訊，請參閱 [《架構部落格》](#) 中的 [欄位備註：使用 FCI 和分散式可用性群組為 SQL Server 建置多區域 AWS 架構](#)。

下圖顯示使用分散式可用性群組的多區域解決方案的範例架構。



## 日誌傳送

日誌運送是一種經過驗證、可靠且符合成本效益的方法，可在發生意外中斷時跨區域保護資料庫。幾十年來，組織一直使用日誌運送來保護其資料。

如果您在實作日誌運送 AWS，您可以根據交易頻率和日誌運送任務，在幾分鐘內達到 RPO 和 RTO。萬一區域變得無法存取，日誌運送會確保您的資料安全且可復原。

請考慮下列使用日誌運送的其他優點：

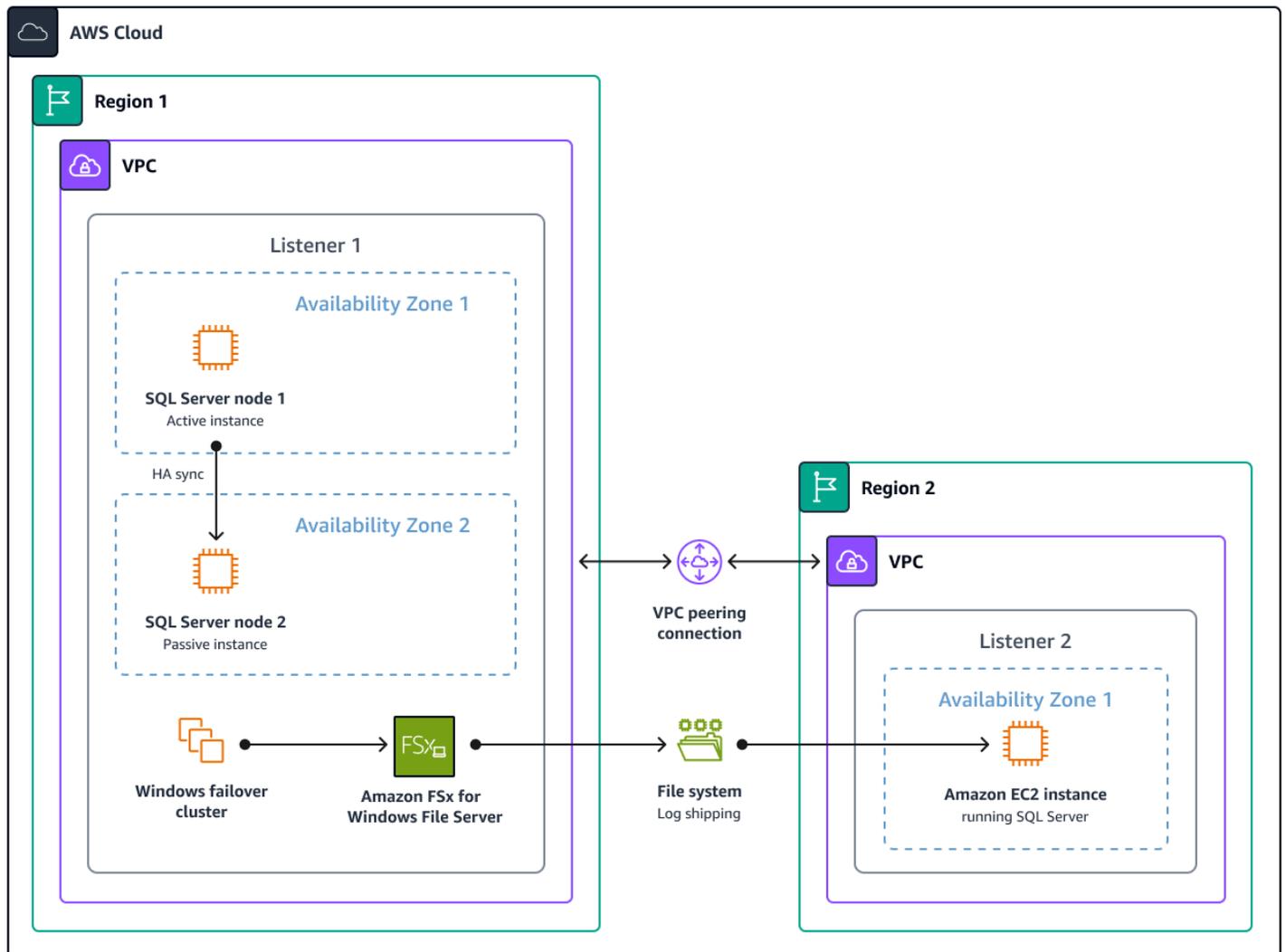
- 透過使用日誌運送來跨區域進行災難復原彈性，以降低成本並滿足您的業務需求。日誌運送可減少您的 TCO，因為您只需要 SQL Server Standard Edition 或 SQL Server Web Edition 授權。
- 使用具有有效 [軟體保證](#) 的日誌運送，從災難復原/被動伺服器移除授權成本。當您搭配軟體保證使用日誌運送時，只需要授權主要/作用中 SQL Server。
- 透過免除 SQL Server Enterprise Edition 在區域之間設定分散式可用性群組的需求，將 SQL Server 授權成本降低 65-75%。您可以使用 SQL Server Standard Edition 和 SQL Server FCIs 結合日誌運送，以滿足災難復原需求。

**Note**

如需 SQL Server 版本之間成本差異的其他資訊，請參閱本指南的[比較 SQL Server 版本](#)一節。

如需詳細資訊，請參閱《AWS 架構部落格》中的[使用 SQL Server FCI 的日誌運送搭配 Amazon FSx for Windows 組態擴展 SQL Server DR](#)。

下圖顯示日誌運送解決方案的範例架構。

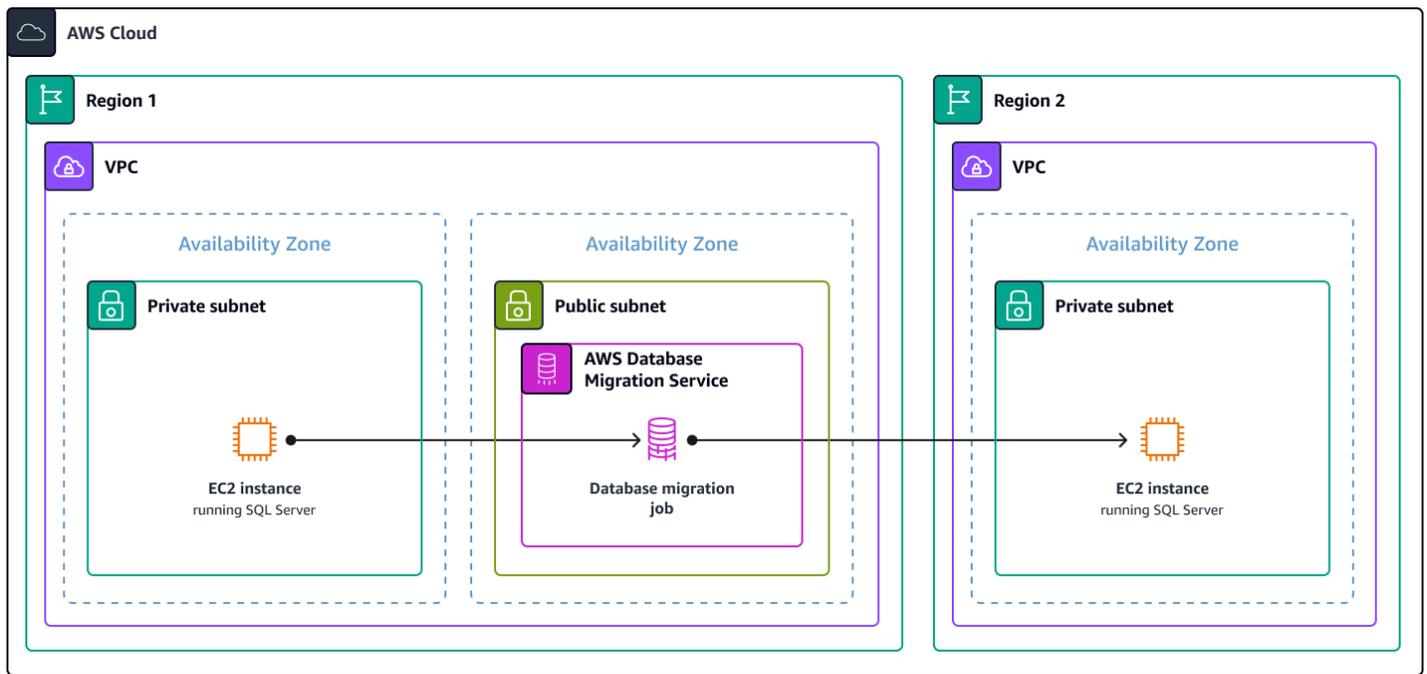


## AWS Database Migration Service

您可以使用 AWS Database Migration Service (AWS DMS) 根據您的應用程式需求設計 HA/DR 解決方案。AWS DMS 可讓您輕鬆地將資料複製到相同區域 (HA) 或跨區域 (DR) 的次要 SQL Server 資料庫。這種方法在技術上健全，可讓您在最佳化資源用量的同時，最大化對 AWS 基礎設施的投資。

AWS DMS 是一項經濟實惠的服務。您只需支付傳輸程序和任何其他日誌儲存期間使用的 CPU 資源的費用。這表示您可以受益於此解決方案，而不會產生重大的額外成本。您可以使用 AWS DMS 來確保您的資料可用且可存取，同時將與授權和資源用量相關的成本降至最低。

下圖顯示以為基礎的解決方案的範例架構 AWS DMS。



## AWS Elastic Disaster Recovery

有些組織必須確保所有關鍵業務應用程式都已制定災難復原計畫。在過去，許多這些組織大量投資於傳統的災難復原解決方案，這需要您預先建置和維護整個重複的基礎設施。這種方法成本高昂、耗時且難以擴展。

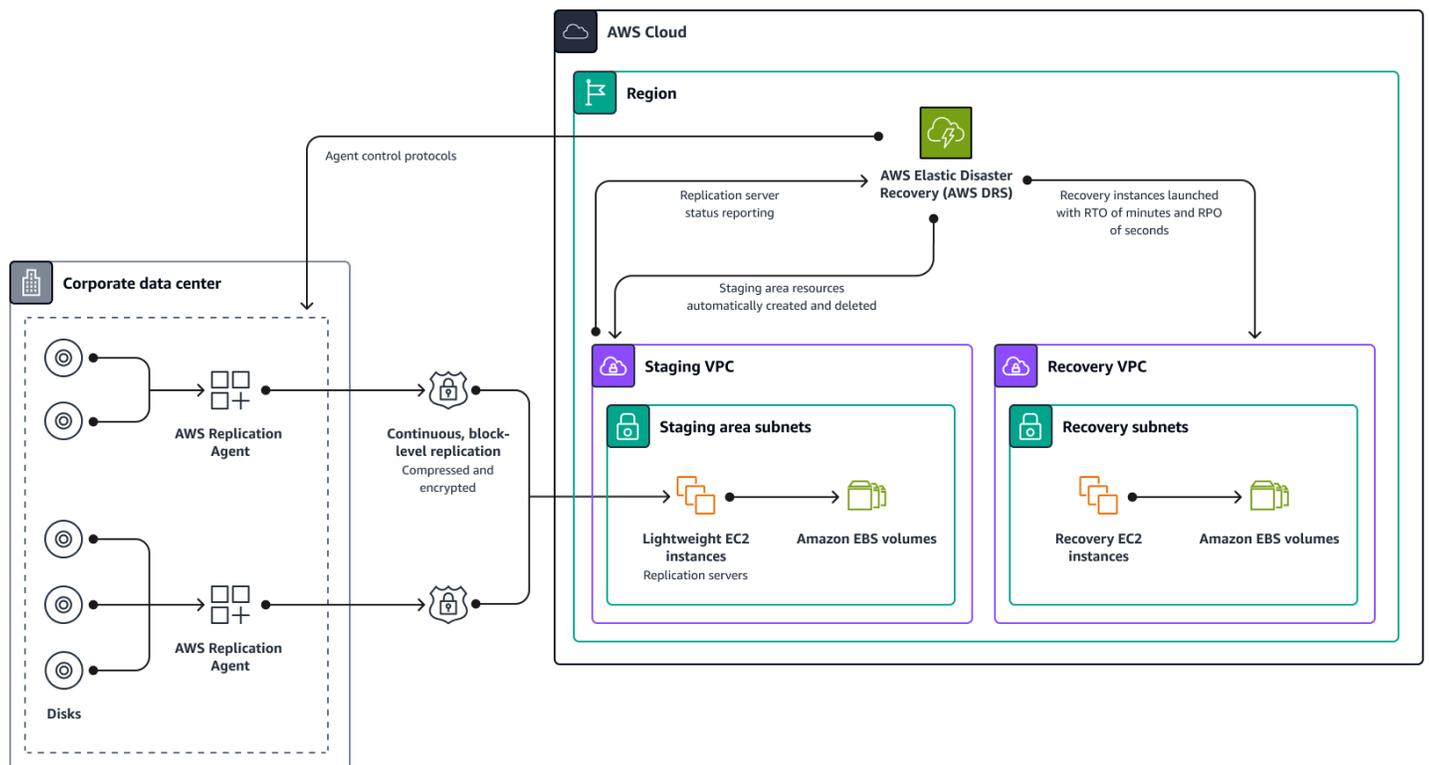
現在，您可以使用 AWS Elastic Disaster Recovery 來消除預先建置災難復原基礎設施的需求。除非必要，否則災難復原機器不會在 Elastic Disaster Recovery 中啟動，因此您只需為在需要時使用的項目付費。這表示您可以大幅降低軟體授權和高效能運算成本。

此外，災難復原解決方案的預備區域包含低成本的 Amazon Elastic Block Store (Amazon EBS) 磁碟區。EBS 磁碟區可進一步降低佈建重複資源的成本。這可讓您降低整體災難復原成本，同時仍維持強

大且可靠的災難復原解決方案，以符合您的業務需求。您可以使用 Elastic Disaster Recovery 來專注於核心業務活動，同時 AWS 負責災難復原解決方案的基礎基礎設施。

對於 SQL Server，您可以使用 Elastic Disaster Recovery 作為經濟實惠的災難復原選項。如果您使用作用中的軟體保證，則會涵蓋容錯、高可用性 SQL Server 架構中被動節點的授權。不過，您仍需支付被動伺服器上線的運算成本。使用 Elastic Disaster Recovery，主要伺服器可以複寫到 DR 環境，而無需維護有效的軟體保證，也無需支付災難復原運算成本。這種節省的組合可以降低 50% 或更多的 SQL Server 災難復原成本。

下圖顯示以 Elastic Disaster Recovery 為基礎的解決方案的範例架構。



如需詳細資訊，請參閱 AWS 部落格上的[如何在 Microsoft 工作負載上使用還原的 DR 網站設定 SQL Server 的高可用性 AWS Elastic Disaster Recovery](#)。

## 成本比較

下表比較本節涵蓋的 HA/DR 解決方案成本。基於此比較的目的，會做出下列假設：

- 執行個體類型 – r5d.xlarge
- 授權類型 – 同時包含 Windows 和 SQL Server 的授權
- 區域 – us-east-1

| 解決方案            | 高可用性 | 災難復原 | 企業版 | 標準版本               | 成本   |
|-----------------|------|------|-----|--------------------|--|
| 日誌傳送            | 否    | 是    | 是   | 是                  | SQL Server Enterprise 版：<br>\$32,674.8<br>(2 個節點)<br><br>SQL Server Standard 版本：<br>\$14,804.4 (2 個節點) |
| Always On 可用性群組 | 是    | 是    | 是   | 是，但基本可用性群組 (2 個節點) | SQL Server Enterprise 版：<br>\$32,674.8<br>(2 個節點)<br><br>SQL Server Standard 版本：<br>\$14,804.4 (2 個節點) |
| Always On FCIs  | 是    | 否    | 是   | 是 (2 個節點)          | SQL Server Standard 版本：<br>14,804.4 美元   |
| 分散式可用性群組        | 是    | 是    | 是   | 否                  | SQL Server Enterprise 版本：<br>\$65,349.6 (4 個節點)  |

| 解決方案   | 高可用性 | 災難復原 | 企業版 | 標準版本 | 成本   |
|--------|------|------|-----|------|--|
| 彈性災難復原 | 否    | 是    | 是   | 是    | <p>近似值。每月 107.48 USD，用於複寫 1 個執行個體和 1 TB 的儲存體</p> <p>注意：彈性災難復原會按每個複寫伺服器每小時計費。無論磁碟數量、儲存體大小、演練或復原啟動次數或您複寫的區域為何，成本都相同。</p> |

| 解決方案             | 高可用性 | 災難復原 | 企業版 | 標準版本 | 成本  |
|------------------|------|------|-----|------|---|
| SIOS Data Keeper | 是    | 是    | 是   | 是    | 具有軟體保證的 Always On 可用性群組 (2 個節點、24 個核心) :<br>213,480 美元<br><br>使用 SIOS DataKeeper 和軟體保證在 SQL Server Standard 版上執行的 2 節點 SQL Server 叢集 : \$61,530 (2 個節點) |
| AWS DMS          | 否    | 是    | 是   | 是    | r5.xlarge 執行個體和 1 TB 儲存體每月<br>745.38 美元   |

## 成本最佳化建議

我們建議您採取下列後續步驟，選擇符合組織需求的 HA/DR 解決方案：

- 檢閱本指南的 [為 SQL Server 工作負載選取正確的 EC2 執行個體](#) 一節。
- 在尖峰工作負載期間執行效能計數器，判斷工作負載的 IOPS 和輸送量需求：
  - IOPS = 磁碟讀取/秒 + 磁碟寫入/秒
  - 輸送量 = 磁碟讀取位元組/秒 + 磁碟寫入位元組/秒
- 使用下列儲存磁碟區類型可提升效能並節省成本：
  - tempdb 和緩衝集區擴充功能的 NVMe 執行個體儲存體

- 資料庫檔案的 io2 磁碟區
- [AWS Trusted Advisor](#) 用於 Amazon EC2 上 SQL Server 成本最佳化的建議。您不需要安裝代理程式 Trusted Advisor 來執行 SQL Server 最佳化檢查。會 Trusted Advisor 檢查您的 Amazon EC2 SQL Server 包含授權的執行個體組態，例如虛擬 CPUs(vCPUs)、版本和版本。然後，根據最佳實務 Trusted Advisor 提出建議。
- 針對 Amazon EC2 執行個體和 Amazon EBS 適當調整大小建議 AWS Compute Optimizer 使用。
- 使用 [AWS 定價計算工具](#) 設計用於成本估算的 HA/DR 策略。
- 若要判斷從 SQL Server Enterprise Edition 降級為 SQL Server Standard Edition 是否為可能的選項，請使用 [sys dm\\_db\\_persisted\\_sku\\_features](#) 動態管理檢視來識別目前資料庫中作用中的版本特定功能。

### Note

使用包含授權的 EC2 執行個體時，SQL Server 版本變更需要 Side-by-side 遷移。

- 執行半年或每年災難復原演練，以更完善地建構設計，以使用定義的 RTO 和 RPO 來復原資料庫。這也可以協助您識別任何架構弱點。

## 其他資源

- [使用 Amazon FSx for Windows File Server 簡化您的 Microsoft SQL Server 高可用性部署](#) (AWS 儲存部落格)
- [欄位備註：使用 FCI 和分散式可用性群組為 SQL Server 建置多區域架構](#) (AWS 架構部落格)
- 在 [上架構 SQL Server 的災難復原 AWS：第 1 部分](#) (AWS 資料庫部落格)
- [Amazon FSx for Windows 的 Microsoft SQL 高可用性](#) (YouTube)
- [使用 Amazon EBS 最大化 Microsoft SQL Server 效能](#) (AWS 儲存部落格)
- [比較您的內部部署儲存模式與 AWS 儲存服務](#) (AWS 儲存部落格)
- [計劃將資料中心 NAS 取代為 Amazon FSx 檔案閘道](#) (AWS 儲存部落格)
- 在 [上最佳化高可用性 SQL Server 部署的成本 AWS](#) (AWS 儲存部落格)
- [如何使用 設定 SQL Server Always On 可用性群組的災難復原 AWS Elastic Disaster Recovery](#) (Microsoft 工作負載開啟 AWS)
- [如何在使用 \(Microsoft 工作負載開啟\) 還原的 DR 網站上設定 SQL Server 的高可用性 AWS Elastic Disaster Recovery](#) AWS

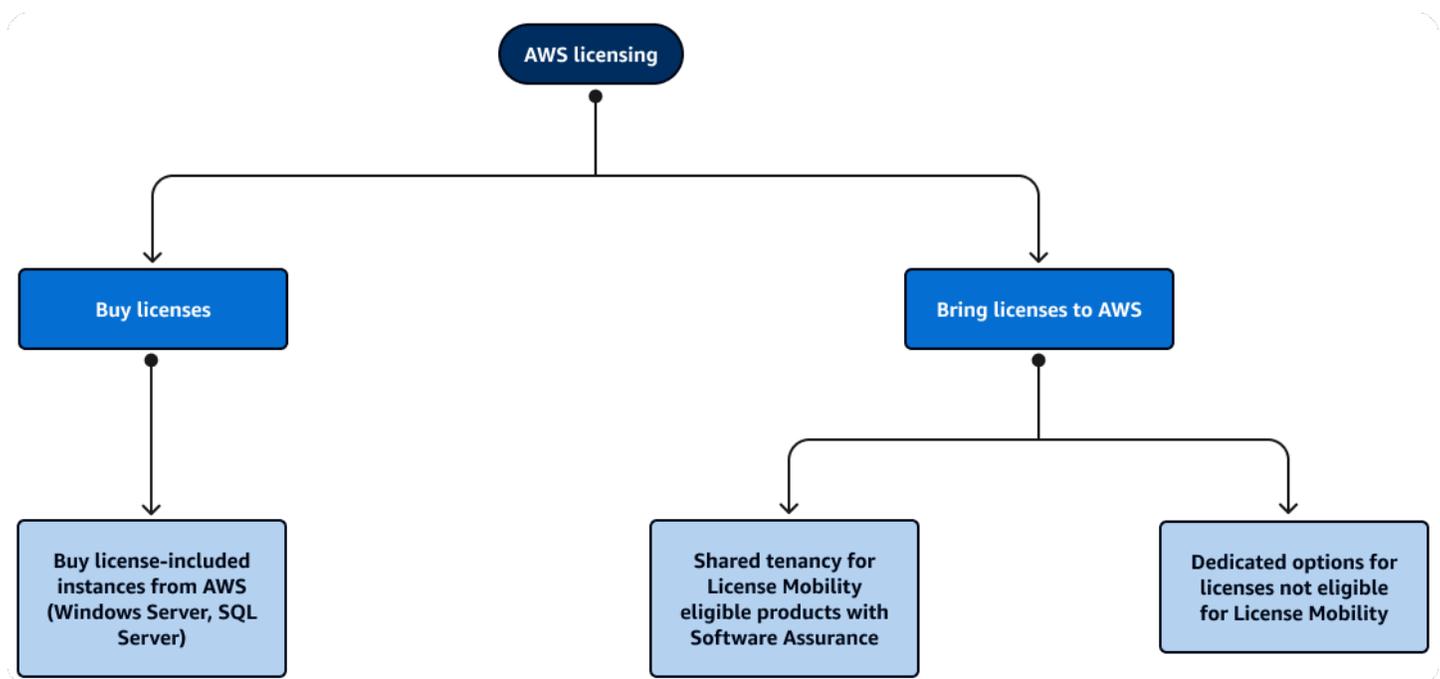
# 了解 SQL Server 授權

## 概觀

隨著越來越多的企業將工作負載移至雲端，雲端平台上的成本最佳化已成為首要任務。授權是與執行 Microsoft 工作負載相關的最重要成本之一 AWS。本節說明如何 AWS 透過最佳化 SQL Server 的 Microsoft 授權來最佳化的成本。

## AWS 授權選項

AWS 為授權提供各種彈性的成本最佳化選擇。這些授權選項旨在協助您降低成本、維持合規性，並滿足您的業務需求。



AWS 將授權分類為三種主要類型：

1. 包含授權 – 此授權選項可讓您隨需購買和使用授權，只需按使用量付費。包含授權的選項非常適合您需要彈性使用授權，並希望避免預付成本的情況。您可以從 Windows Server、SQL Server 和其他 Microsoft 產品中選擇。
2. 使用自有授權 (BYOL) 產品搭配授權行動性 – 此授權選項適用於您已擁有現有授權並希望在雲端使用這些授權的案例。AWS 可讓客戶透過 Microsoft 的授權[行動性](#)計畫，將自己的授權帶到雲端。您可以將具有授權行動性的產品，例如具有軟體保證 (SA) 的 SQL Server，帶到共用或專用租用，以降低 AWS 執行個體成本。

3. 沒有授權行動性的 BYOL 產品 – 對於沒有授權行動性的 Microsoft 產品，例如 Windows Server，AWS 提供在雲端使用這些產品的專用選項。此外，專用主機提供在實體核心層級授權的機會。這可為您節省 50% 或更多執行工作負載所需的授權。對於大多數時間執行的穩定且可預測的工作負載，專用主機是很好的選擇。

## 取得授權的成本影響

使用授權可能會對執行 Microsoft 工作負載的成本產生重大影響 AWS。如果您使用自己的授權，您不需要為在雲端中執行的執行個體支付額外的授權成本。這可能會導致大幅節省成本。

下列比較顯示全年無休執行單一 c5.xlarge 執行個體的隨需每月成本：

- Windows Server + SQL Server Enterprise Edition：每月 1353 美元（包含授權）
- Windows Server + SQL Server Standard 版：\$609/月（包含授權）
- 僅限 Windows Server：每月 259 美元（包含授權）
- 僅限運算 (Linux)：每月 127 美元

最後，使用您自己的授權可能會對執行 Microsoft 工作負載的成本產生重大影響 AWS。如果您使用現有的授權，您可以降低授權成本，並節省整體 AWS 帳單的費用。

## 授權最佳化

最佳化和授權評估 AWS (AWS OLA) 可以降低運算和授權成本，協助您最佳化授權。AWS OLA 旨在評估您在遷移計劃之工作負載上執行的工作負載 AWS 或工作負載的授權需求。AWS OLA 提供最佳化授權使用的建議。

最佳化授權用量的關鍵策略之一是[正確調整執行個體的大小](#)。適當調整大小包括根據工作負載的 CPU、記憶體和儲存需求，選擇適合的執行個體類型。透過選擇適當的執行個體大小，您可以確保以經濟實惠的方式使用資源。這可能會導致大幅節省成本。

使用 Microsoft 軟體授權，軟體執行的核心數量是決定授權成本的關鍵因素。例如，Windows Server 和 SQL Server 授權通常由核心數量授權。透過正確調整執行個體的大小，您可以降低 Microsoft 軟體執行的核心數量，進而同時降低執行個體的成本和所需的授權數量。

## 成本最佳化建議

最佳化授權是成本最佳化的關鍵元件 AWS。透過實作正確的策略，您可以降低授權成本、維持合規性，並從授權投資中獲得最佳價值。本節概述多種授權最佳化策略。

## 攜帶您的合格 Windows Server 授權

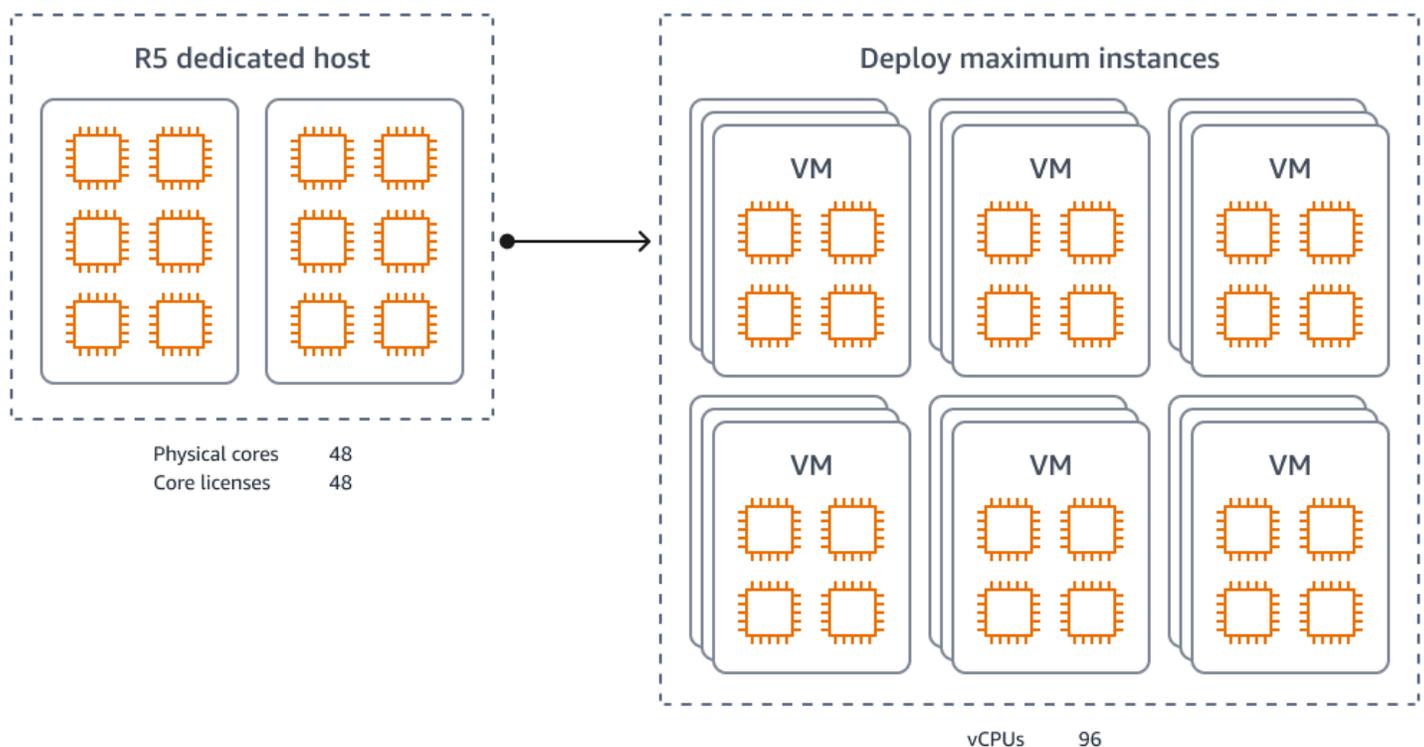
使用您自己的 Windows Server 授權是授權最佳化的最有效策略之一。此策略可讓您利用現有的投資來減少 AWS 支出。

例如，如果您在 1/10/2019 之前購買授權，或根據在該日期之前簽署的作用中 Enterprise Agreement 以校正形式購買授權，則可以在 [Amazon EC2 專用主機](#) 上部署 Windows Server 2019 和舊版。此規則是根據 Microsoft 在 2019 年針對沒有授權行動性之產品的授權條款及條件所做的變更，例如 Windows Server，當部署在 [列出的供應商](#)（例如 Alibaba AWS 或 Google Cloud）上時。根據新條款，您無法將自己的 Windows Server 授權帶到 AWS，但必須改用包含授權的執行個體。不過，如果您在該日期之前購買永久授權，您仍然可以在 Amazon EC2 專用主機上部署這些 Windows Server 授權。

## 實體層級授權

實體核心層級的授權可讓您僅授權主機的實體核心，因此您可以部署最多數量的執行個體，而不會影響所需的授權數量。這通常透過使用 Windows Server Datacenter 和 SQL Server Enterprise Edition 來完成。

例如，請考慮具有 48 個核心的 R5 專用主機，這會轉換為 96 vCPUs。如果您使用 Windows Server Datacenter 版本，則只需要 48 個授權。這可讓您部署最多 96 vCPUs 的執行個體組合，如下圖所示。



如果您有足夠的工作負載來最大化可在主機上執行的執行個體數量，這種方法可能特別符合成本效益。透過在實體核心層級進行授權，您可以避免每個執行個體的額外授權成本，並實現授權投資的最佳價值。

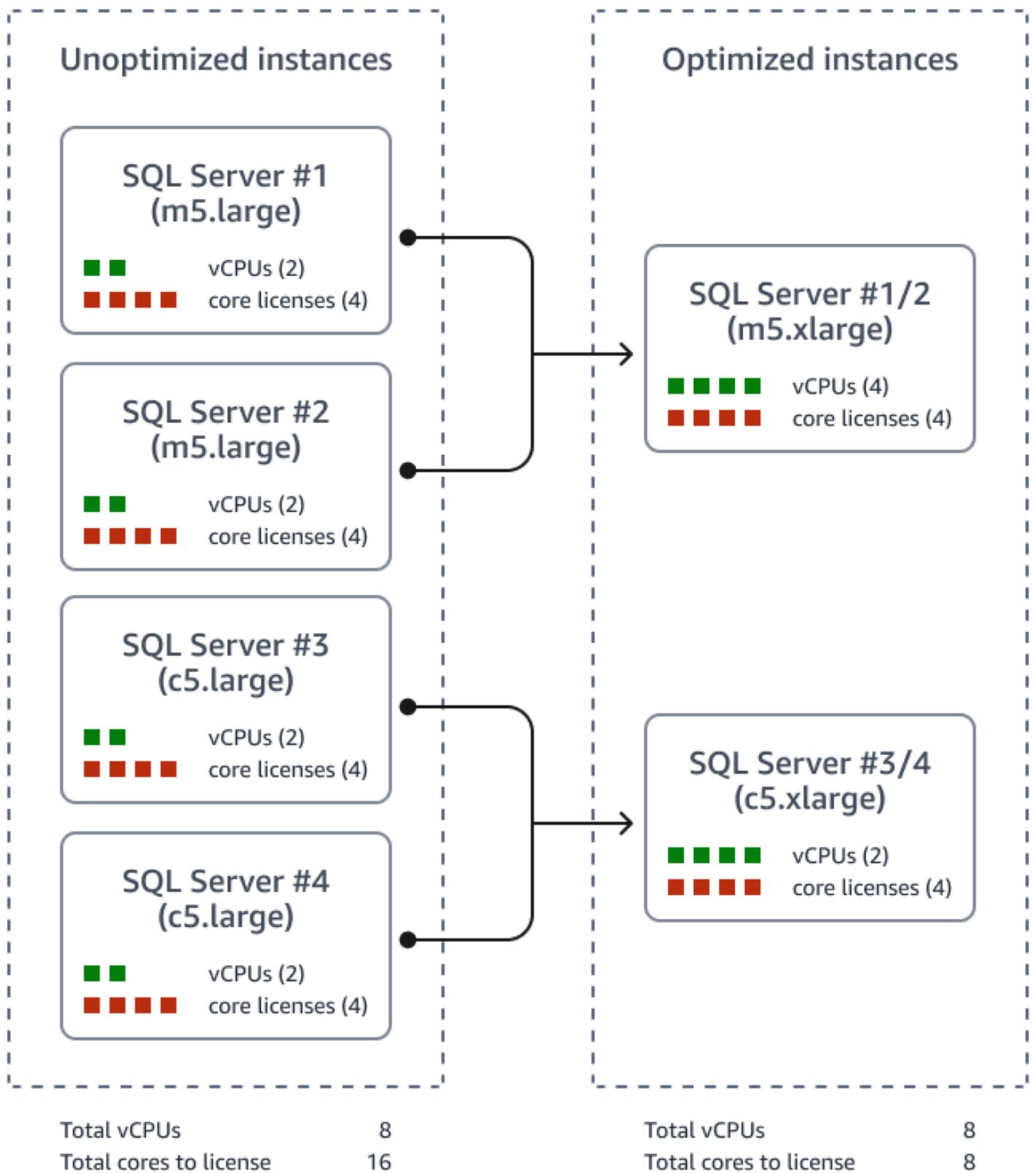
## SQL Server 實體核心層級的授權

在共用租用中，SQL Server 授權是以配置給執行個體的 vCPUs 數目為基礎。相反地，使用專用主機，您可以在實體核心層級或 vCPU 層級授權 SQL Server Enterprise Edition。

如同先前的 R5 專用主機範例，如果您在實體核心層級授權 SQL Server Enterprise Edition，則只需要 48 個 SQL Server Enterprise Edition 授權即可授權主機。相反地，在共用租用中，只有 vCPU 授權的選項，您必須擁有相同工作負載的 96 個 SQL Server Enterprise Edition 授權。因此，相較於共用租用，專用主機最多可為您節省 50% 的 SQL Server 授權成本。這是除了透過使用合格的 Windows 授權來節省執行個體成本之外。

## 合併 SQL Server 執行個體

[SQL Server 合併](#)是將多個 SQL Server 執行個體合併到一個伺服器的程序。SQL Server 每個執行個體至少需要四個核心授權，即使執行個體只有兩個 vCPUs。這表示在少於四個核心的伺服器上執行 SQL Server 可能會導致您過度授權這些執行個體，並在必要時使用更多授權。



例如，將具有兩個 vCPUs 兩個執行個體合併為具有四個 vCPUs 單一執行個體，可將授權需求降低 50%。這是因為只需要四個核心授權，而不是八個。

如需整合的詳細資訊，請參閱本指南的 [SQL Server 整合](#) 一節。

## 降級 SQL Server 版本

[變更 SQL Server 版本](#) 可以是最佳化授權用量和降低成本的關鍵策略。企業版 SQL Server 比標準版更昂貴，因此降級可以大幅節省成本。

透明資料加密 (TDE) 和 Always On 可用性群組是 SQL Server Enterprise Edition 中的兩個熱門功能。不過，如果您不需要完整的 SQL Server Enterprise Edition 功能集，您可以考慮這些功能的成本效益替代方案。例如，您可以從 SQL Server 2019 開始取得 SQL Server Standard 版中的 TDE。取代 Always On 可用性群組，您可以在 FSx for Windows File Server 上使用容錯移轉叢集與共用儲存，以獲得 SQL Server Standard Edition 的高可用性。

透過從 SQL Server Enterprise Edition 降級至 SQL Server Standard Edition，您可以大幅降低授權成本。如需詳細資訊，請參閱 AWS Storage Blog [上的 上高可用性 SQL Server 部署的最佳化成本 AWS](#)。

除了降低授權成本之外，降級 SQL Server 版本也有助於降低您的軟體保證支出，並協助您避免未來的校正。如果您將未使用的授權退回到機架，您可以避免額外的授權成本，並從授權投資中獲得最佳價值。

請務必仔細評估 SQL Server 工作負載，並判斷哪些功能對您的業務需求至關重要。如需詳細資訊，請參閱 AWS Prescriptive Guidance [中的評估您的環境](#)，並判斷您的 Microsoft SQL Server 資料庫是否使用 SQL Server Enterprise Edition 特定功能。

如果您選擇正確的 SQL Server 版本，並使用 SQL Server Enterprise Edition 功能的替代方案，您可以大幅節省成本，同時保持合規性並滿足您的業務需求。如需降級選項的詳細資訊，請參閱本指南的 [比較 SQL Server 版本](#) 一節。

## 在非生產環境中使用 SQL Server Developer Edition

在非生產環境中，您可以在內部部署環境中使用 MSDN 訂閱來部署 SQL Server 的可授權版本，例如 Enterprise 或 Standard Edition。不過，MSDN 訂閱沒有授權行動性。因此，如果您遷移到 AWS，則無法將這些授權帶過來。您必須改用 SQL Server Developer Edition。

SQL Server Developer Edition 是全功能版的 SQL Server，可免費使用。此版本適用於 SQL Server 2016 版及更新版本。您可以從 Microsoft 網站下載。SQL Server Developer Edition 旨在用於所有非生產環境，例如開發、測試和預備，只要它未連接到即時生產資料。

如果您在非生產環境中使用 SQL Server Developer Edition，您可以避免額外的授權成本。如需詳細資訊，請參閱本指南的[評估 SQL Server 開發人員版本](#)一節。

## 最佳化 SQL Server 工作負載的 CPU

在某些情況下，由於 RAM 或聯網限制等其他因素，您可能需要選擇 CPU 超過工作負載所需 CPUs 的執行個體類型。不過，AWS 提供解決方案，協助您在這些情況下最佳化授權成本。

您可以像大多數使用 SQL Server 核心授權的客戶一樣，停用超執行緒或關閉 EC2 執行個體上的 CPUs，以限制主機的可用 CPUs 數量。此選項可讓您利用 RAM 等其他執行個體功能，同時節省購買額外授權的成本。

例如，如果您部署 r5.4xlarge 執行個體，因為您的工作負載需要 128 GB 的記憶體，但只需要八個 SQL Server 核心，則您可以在執行個體啟動時停用超執行緒，只需要八個作用中 CPUs。透過這樣做，您可以節省 50% 的必要 SQL Server 授權，因為您只需要授權目前正在使用的八個核心。

| 執行個體類型      | vCPUs總數 | 具有最佳化 CPUs功能的作用中 vCPU | SQL Server 授權節省 |
|-------------|---------|-----------------------|-----------------|
| r5.4xlarge  | 16      | 8                     | 50%             |
| r5.12xlarge | 48      | 8                     | 83%             |

如果您授權執行個體，您可以確保為工作負載使用最具成本效益的執行個體類型。隨著 AWS 推出新的執行個體類型，請務必評估這些新執行個體是否能夠以較少的核心滿足工作負載需求。

## 其他資源

- [Amazon Web Services 和 Microsoft：常見問答集](#) (AWS 文件)

## 為 SQL Server 工作負載選取正確的 EC2 執行個體

### Important

閱讀本節之前，建議您先閱讀本指南的[了解 SQL Server 授權](#)和[為 Windows 工作負載選取正確的執行個體類型](#)一節。

## 概觀

Microsoft SQL Server 已在 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體上執行超過 15 年。AWS 已採用該經驗並用它來協助開發 Amazon EC2 執行個體，以適應從最低規格到高效能多區域叢集執行的 SQL Server 工作負載。

選擇 SQL Server 的正確 EC2 執行個體，在很大程度上取決於您的工作負載。了解 SQL Server 如何獲得授權、如何使用記憶體，以及 SQL Server 功能如何與 Amazon EC2 產品保持一致，有助於引導您找到應用程式的最佳 EC2 執行個體。

本節說明各種 SQL Server 工作負載，以及如何與特定 EC2 執行個體配對，將您的授權和運算成本降至最低。

## 成本比較

Amazon EC2 可讓您使用自有授權 (BYOL) 或使用 Windows Server 和 SQL Server 授權付費。對於隨 pay-as-you-go 授權，Windows Server 和 SQL Server 授權的授權成本會按 EC2 執行個體的每小時成本計算。例如，您可以擁有價格不同的不同 AMIs。AMI 的價格取決於 AMI 執行所在的 SQL Server 版本。

Windows Server 和 SQL Server 定價未逐項列出。您在 [等工具](#) 上找不到明細定價 [AWS 定價計算工具](#)。如果您選擇包含授權方案的不同組合，則可以降低授權成本，如下表所示。

| EC2 執行個體  | AMI                 | 運算價格      | Windows 授權價格 | SQL 授權價格 | 總價        |
|-----------|---------------------|-----------|--------------|----------|-----------|
| r5.xlarge | Linux ( 運算定價 )      | 183.96 美元 | -            | -        | 183.96 美元 |
| r5.xlarge | Linux + SQL 開發人員    | 183.96 美元 | 0 USD        | 0 USD    | 183.96 美元 |
| r5.xlarge | Windows Server (LI) | 183.96 美元 | 134.32 美元    | -        | 318.28 美元 |
| r5.xlarge | Windows + SQL 開發人員  | 183.96 美元 | 134.32 美元    | 0 USD    | 318.28 美元 |

| EC2 執行個體  | AMI                           | 運算價格      | Windows 授權價格 | SQL 授權價格 | 總價         |
|-----------|-------------------------------|-----------|--------------|----------|------------|
| r5.xlarge | Windows + SQL Web (LI)        | 183.96 美元 | 134.32 美元    | 49.64 美元 | 367.92 美元  |
| r5.xlarge | Windows + SQL 標準 (LI)         | 183.96 美元 | 134.32 美元    | 350.4 美元 | 668.68 美元  |
| r5.xlarge | Windows + SQL Enterprise (LI) | 183.96 美元 | 134.32 美元    | 1095 美元  | 1413.28 美元 |

### Note

上表中的定價是以 us-east-1 區域中的隨需定價為基礎。

執行 SQL Server 最符合成本效益的方法是保持在較低層級的版本，直到您需要更高層級版本的功能為止。如需詳細資訊，請參閱本指南的[比較 SQL Server 版本](#)一節。從 SQL Server Web Edition 升級至 SQL Server Standard Edition 的費用超過 SQL Server 授權成本的七倍，以及從 Standard Edition 移至 Enterprise Edition 的費用超過三倍。授權成本的差異是需要考慮的主要因素，並在本節的其餘部分中探索。

## 成本最佳化案例

請考慮分析公司追蹤交付工具正在尋求改善 SQL Server 效能的範例案例。MACO 專家審查公司的效能瓶頸後，公司會從 x1e.2xlarge 執行個體轉換為 x2iedn.xlarge 執行個體。雖然執行個體大小較小，但 x2 執行個體的增強功能會使用緩衝集區延伸來改善 SQL Server 效能和最佳化。這可讓公司從 SQL Server Enterprise Edition 降級為 SQL Server Standard Edition，並將 SQL Server 授權從 8 個 vCPUs 減少為 4 vCPUs。

最佳化之前：

| Server  | EC2 執行個體    | SQL Server 版本 | 每月成本        |
|---------|-------------|---------------|-------------|
| ProdDB1 | x1e.2xlarge | Enterprise    | 3,918.64 美元 |
| ProdDB2 | x1e.2xlarge | Enterprise    | 3,918.64 美元 |
| 總計      |             |               | 7,837.28 美元 |

最佳化後：

| Server  | EC2 執行個體      | SQL Server 版本 | 每月成本        |
|---------|---------------|---------------|-------------|
| ProdDB1 | x2iedn.xlarge | 標準            | 1,215.00 美元 |
| ProdDB2 | x2iedn.xlarge | 標準            | 1,215.00 美元 |
| 總計      |               |               | 2,430.00 美元 |

從 x1e.2xlarge 執行個體到 x2iedn.xlarge 執行個體的合併變更，可讓範例客戶在生產資料庫伺服器上每月節省 5,407 美元。這可將工作負載的總成本降低 69%。

#### Note

上表中的定價是以 us-east-1 區域中的隨需定價為基礎。

## 成本最佳化建議

### 記憶體最佳化執行個體

SQL Server 最重要的層面之一是了解其對記憶體的依賴。SQL Server 會嘗試使用作業系統未使用的所有可用 RAM（預設安裝最多 2 TB）。它出於效能原因這樣做。與必須持續從磁碟提取資料、進行變更，然後將其寫入磁碟相比，在記憶體中使用資料的效能更高。反之，SQL Server 會嘗試盡可能從連接的資料庫載入資料，並將該資料保留在 RAM 中。對資料所做的變更會在記憶體中發生，並在稍後強化為磁碟。

**Note**

如需 SQL Server 如何寫入變更的詳細說明，請參閱 Microsoft 文件中的[寫入頁面](#)。

由於 SQL Server 在大量 RAM 中表現較佳，因此通常建議從 [Amazon EC2 記憶體最佳化](#) 執行個體類型開始。記憶體最佳化執行個體用途廣泛，提供各種不同的選項。R 系列具有 1 比 8 的 vCPU-to-RAM 比率，並可選擇 Intel 處理器、AMD 處理器、增強型聯網、增強型 EBS 效能、執行個體儲存和增強型處理器速度。對於記憶體密集型工作負載，也有一個 X 系列，結合了許多相同的選項，並將 vCPU-to-RAM 的比率擴展到 1 與 32。由於記憶體最佳化執行個體的多樣性，您可以將它們套用至所有形狀和大小的 SQL Server 工作負載。

### 低於最小資源的工作負載（少於 4vCPUs）

雖然某些使用案例適用於爆量 (T3) 執行個體，但我們建議您通常避免將爆量執行個體用於 SQL Server 工作負載。SQL Server 的授權是根據指派給執行個體 vCPUs 數量。如果 SQL Server 在一天的大部分時間處於閒置狀態並取得爆量額度，則您需要為未充分利用的 SQL 授權付費。此外，SQL Server 具有每個伺服器 4 個核心的最低授權需求。這表示如果您的 SQL Server 工作負載不需要 4 vCPUs 的運算能力，則需要支付您未使用的 SQL Server 授權。在這些情況下，最好將 [多個 SQL Server 執行個體](#) 合併到較大的伺服器上。

### 使用最少資源的工作負載（小於 64 GB RAM）

許多 64 GB RAM 以下的 SQL Server 工作負載不會優先考慮高效能或高可用性。對於這些類型的工作負載，如果應用程式涵蓋在 Microsoft 的授權限制中，則 SQL Server Web 版本可能非常適合。

**Important**

SQL Server Web Edition 具有以 Microsoft 授權條款為基礎的限制使用案例。SQL Server Web 版可能僅支援公有和網際網路可存取的網頁、網站、Web 應用程式和 Web 服務。它可能不會用於支援企業營運應用程式 (例如，客戶關係管理、企業資源管理和其他類似的應用程式)。

SQL Server Web 版可擴展到 32 個 vCPUs 和 64 GB RAM，比 SQL Server Standard 版便宜 86%。對於低資源工作負載，使用 r6a 等 AMD 記憶體最佳化執行個體，其運算價格比 Intel 便宜 10%，也是將運算和 SQL 授權成本降至最低的好方法。

## 平均資源的工作負載 ( 小於 128 GB RAM)

SQL Server Standard Edition 用於大多數高達 128 GB RAM 的 SQL Server 工作負載。SQL Server Standard 版比 SQL Server Enterprise Edition 便宜 65-75%，可擴展至 48 個 vCPUs 和 128 GB RAM。由於 128 GB RAM 限制通常會在 48 個 vCPU 限制之前達到，因此大多數想要避免升級至 SQL Server Enterprise Edition 的客戶都會成為重點。

SQL Server 具有稱為[緩衝集區延伸](#)的功能。此功能可讓 SQL Server 使用磁碟的一部分做為 RAM 的延伸。緩衝集區擴充功能與超快速儲存結合時運作良好，例如 [Amazon EC2 執行個體儲存](#) 中使用的 NVMe SSDs。包含執行個體儲存體的 Amazon EC2 執行個體在執行個體名稱中以「d」表示 ( 例如 r5d、r6id 和 x2iedn)。

緩衝集區擴充功能無法取代一般 RAM。不過，如果您需要超過 128 GB 的 RAM，您可以搭配 r6id.4xlarge 和 x2iedn.xlarge 等 EC2 執行個體使用緩衝集區延伸，以延遲升級至 Enterprise Edition 授權。

## 高效能工作負載 ( 超過 128 GB RAM)

需要高效能的 SQL Server 工作負載在成本最佳化方面具有挑戰性，因為它們依賴許多資源。不過，了解 EC2 執行個體的差異可能會讓您無法做出錯誤的選擇。

下表顯示各種記憶體最佳化的 EC2 執行個體及其效能限制。

|              | r5b                           | r6idn                         | r7iz                             | x2iedn                        | x2iezn                        |
|--------------|-------------------------------|-------------------------------|----------------------------------|-------------------------------|-------------------------------|
| 處理器          | 3.1 GHz<br>第二代 Intel Xeon 處理器 | 3.5 GHz<br>第三代 Intel Xeon 處理器 | 3.9 GHz<br>第四代 Intel Xeon 可擴充處理器 | 3.5 GHz<br>第三代 Intel Xeon 處理器 | 4.5 GHz<br>第二代 Intel Xeon 處理器 |
| CPU : RAM 比率 | 1 : 8                         | 1 : 8                         | 1 : 8                            | 1 : 32                        | 1 : 32                        |
| 最大 vCPU      | 96                            | 128                           | 128                              | 128                           | 48                            |
| 最大 RAM       | 768 GB                        | 1,024 GB                      | 1,024 GB                         | 4,096 GB                      | 1,536 GB                      |
| 執行個體儲存體      | –                             | NVMe SSD                      | –                                | NVMe SSD                      | –                             |

|                   | r5b     | r6idn        | r7iz    | x2iedn       | x2iezn   |
|-------------------|---------|--------------|---------|--------------|----------|
|                   |         | (4x 1900 GB) |         | (2x 1900 GB) |          |
| io2 Block Express | 支援      | 支援           | 支援      | 支援           | –        |
| 最大 EBS IOPS       | 260,000 | 350,000      | 160,000 | 260,000      | 80,000   |
| EBS 輸送量上限         | 60 Gbps | 80 Gbps      | 40Gbps  | 80 Gbps      | 19 Gbps  |
| 網路頻寬上限            | 25 Gbps | 200 Gbps     | 50 Gbps | 100 Gbps     | 100 Gbps |

每個執行個體都用於不同的用途。了解 SQL Server 工作負載可協助您選擇最適合您的執行個體類型。

屬性的詳細資訊：

- r5b – r5b 中的「b」屬性表示此執行個體類型著重於高 EBS 效能。在第五代記憶體最佳化執行個體中，r5b 是首選。這是第一個使用 io2 Block Express 磁碟區的執行個體類型，並達到 260,000 的最大儲存 IOPS。r5b 執行個體類型仍然是符合成本效益的替代方案，可滿足高 EBS 效能需求。
- r6idn – 第六代的記憶體最佳化執行個體提供比上一代大幅改善。r5b 的 EBS 效能增強功能會進一步採用 r6idn，將 IOPS 上限增加到 350,000。r6idn 也有 tempdb 和緩衝集區擴充功能的執行個體存放區磁碟區，可進一步提升 SQL Server 效能。
- x2iedn – x2iedn 類似於 r6idn。它提供類似層級的增強型 EBS、增強型聯網和 NVMe SSD 執行個體儲存體，但高記憶體工作負載和低 CPU 數量（較低的 SQL Server 授權成本）的 1:32 vCPU-to-RAM 比率。
- x2iezn – x2iezn 中的 "z" 屬性表示此執行個體類型著重於高處理器效能。Cascade Lake 處理器具有高達 4.5 GHz 的全核心渦輪頻率。在您想要保持低 vCPU 數量的情況下，我們建議您使用此 EC2 執行個體，並搭配 1:32 vCPU 與 RAM 的比率。vCPU-to-RAM 這反過來可以降低 SQL Server 授權成本。
- r7iz – r7iz 中的 "z" 屬性表示此執行個體類型著重於高處理器效能。Sapphire Fast 處理器具有高達 3.9 GHz 的全核心渦輪頻率。如同 x2iezn 執行個體，r7iz 會優先考慮高頻率處理器效能，但採用 1:8 vCPU-to-RAM 的比率。

## 其他資源

- [一般用途 Amazon EC2 執行個體](#) (AWS 文件)
- [比較工具](#) (Vantage)
- [授權 – SQL Server](#) (AWS 文件)

## 合併執行個體

本節著重於將多個 SQL Server 執行個體合併到相同伺服器的成本最佳化技術，以將授權成本降至最低，並最大化資源使用率。

### 概觀

建立執行個體是安裝 SQL Server 資料庫引擎程序的一部分。SQL Server 執行個體是完整的安裝，包含自己的伺服器檔案、安全登入和系統資料庫（主要、模型、msdb 和 tempdb）。由於執行個體擁有自己的所有檔案和服務，因此您可以在相同的作業系統上安裝多個 SQL Server 執行個體，而不需要執行個體互相干擾。不過，由於執行個體都安裝在相同的伺服器上，因此它們都共用相同的硬體資源，例如運算、記憶體和聯網。

在生產環境中，通常每個伺服器僅使用單一 SQL Server 執行個體，因此「忙碌」執行個體不會過度使用共用硬體資源。為每個 SQL Server 執行個體提供自己的作業系統和自己的資源，比倚賴資源控管來得好。對於需要大量 RAM 和 CPU 資源的高效能 SQL Server 工作負載尤其如此。

不過，並非所有 SQL Server 工作負載都會使用大量資源。例如，為了合規或安全目的，某些組織會為其每位客戶指派自己的專用 SQL Server 執行個體。對於較小型的用戶端或通常非作用中的用戶端，這表示以最少的資源執行 SQL Server 執行個體。

如 [Microsoft SQL Server 2019：授權指南](#) 中所述，每個執行 SQL Server 的伺服器必須至少考慮四個 CPU 授權。這表示即使您執行的伺服器只有兩個 vCPUs，您仍然必須為四個 vCPUs 授權 SQL Server。如果您使用 [SQL Server Standard 版](#)，則根據 [Microsoft 的公有 SQL Server 定價](#) 差異為 \$3,945。對於使用最少資源搭配單一 SQL Server 執行個體執行多個伺服器的組織，必須授權未使用資源的合併成本可能相當龐大。

## 成本最佳化案例

本節探討的範例案例會比較執行四個 Windows Server 伺服器，每個伺服器與單一 SQL Server 執行個體，以及同時執行多個 SQL Server 執行個體的單一大型 Windows Server 伺服器之間的差異。

如果每個 SQL Server 執行個體只需要兩個 vCPUs 和 8 GB RAM，除了每小時運算成本 0.096 美元之外，每部伺服器的總成本為 SQL Server 授權 7,890 美元。

| EC2 執行個體  | vCPU | RAM | 價格    | 要授權 vCPUs | SQL Server 授權總成本 |
|-----------|------|-----|-------|-----------|------------------|
| m6i.large | 2    | 8   | 0.096 | 4         | 7,890 美元         |

將此擴展到四個伺服器，SQL Server 授權的總成本為 31,560 USD，每小時運算成本為 0.384 USD。

| EC2 執行個體         | vCPU | RAM | 價格    | 要授權 vCPUs | SQL Server 授權總成本 |
|------------------|------|-----|-------|-----------|------------------|
| 4 倍<br>m6i.large | 2    | 32  | 0.384 | 16        | 31,560 美元        |

如果您將所有四個 SQL Server 執行個體合併到單一 EC2 執行個體，則運算資源和運算的總量會保持不變。不過，透過移除不必要的 SQL Server 授權成本，您可以將執行工作負載的總成本降低 15,780 美元。

| EC2 執行個體    | vCPU | RAM | 價格    | 要授權 vCPUs | SQL Server 授權總成本 |
|-------------|------|-----|-------|-----------|------------------|
| m6i.2xlarge | 8    | 32  | 0.384 | 8         | 15,780 美元        |

#### Note

在上述表格中，運算成本會顯示 us-east-1 區域中執行 Windows Server 的 Amazon EC2 伺服器每小時隨需定價。SQL Server Standard Edition 授權成本是指 [Microsoft 的公有 SQL Server 定價](#)。

## 成本最佳化建議

如果您正在考慮合併 SQL Server 執行個體，最大的考量是您要合併的每個執行個體的資源耗用。請務必長期取得效能指標，以便更深入了解每個伺服器上的工作負載模式。資源耗用量監控的一些常用工具包括 [Amazon CloudWatch](#)、[Windows Performance Monitor](#) (perfmon) 和 SQL Server 的 [原生監控工具](#)。

我們建議您在分析 SQL Server 工作負載是否可以合併以使用相同的伺服器資源，而不會互相干擾時，考慮下列問題：

- 穩定狀態期間會耗用哪些資源 (CPU、記憶體和網路頻寬) ？
- 尖峰期間會耗用哪些資源 (CPU、記憶體和網路頻寬) ？
- 峰值發生的頻率為何？ 峰值是否一致？
- 一個伺服器的資源峰值是否與另一個伺服器的資源峰值一致？
- SQL Server 使用的儲存 [IOPS](#) 和 [輸送量](#) 是多少？

如果您想要繼續進行合併 SQL Server 執行個體的計劃，請參閱 Cloud Operations & Migrations 部落格上的在一個 [Amazon EC2 執行個體上執行多個 SQL Server 執行個體](#) 文章。AWS 此文章提供如何在 SQL Server 中進行組態變更以新增其他執行個體的指示。開始之前，請考慮在相同伺服器上安裝多個執行個體時的次要差異：

- 預設 SQL Server 資料庫執行個體名為 MSSQLSERVER，並使用連接埠 1433。
- 安裝在相同伺服器上的每個額外執行個體都是「已命名」的資料庫執行個體。
- 每個具名執行個體都有唯一的執行個體名稱和唯一的連接埠。
- [SQL Server 瀏覽器](#) 必須執行來協調指定執行個體的流量。
- 每個執行個體都可以針對資料庫資料檔案和個別登入使用不同的位置。
- SQL Server [最大伺服器記憶體設定](#) 必須根據每個執行個體的效能需求進行設定，其總和也為基礎作業系統保留足夠的記憶體。
- 您可以使用 SQL Server [原生備份和還原](#) 功能，或 [AWS DMS](#) 用於遷移或整合。

## 其他資源

- [SQL Server 授權資料表](#) (AWS 雲端操作與遷移部落格)
- [SQL Server 多執行個體設定部落格文章](#) (AWS 雲端操作與遷移部落格)
- [SQL Server 最佳實務指南](#) (AWS 方案指引文件)

# 比較 SQL Server 版本

## 概觀

Microsoft SQL Server 授權是 Windows 工作負載環境的最大費用之一。SQL Server 的授權成本可以輕鬆擴展到超過執行工作負載的運算成本。如果您選擇了錯誤的版本，您可以支付未使用或甚至不需要的功能。本節會比較下列 SQL Server 版本，包括其功能和相對成本：

- 企業 – SQL Server Enterprise Edition 為資料中心提供高效能、無限制的虛擬化和數種商業智慧 (BI) 工具。
- 標準 – SQL Server Standard Edition 為小型組織和部門提供基本資料管理和商業智慧。
- Web – SQL Server Web 版適用於 Web 主機或 Web 增值提供者 (VAPs) 的公司。此版本提供低總擁有成本，並為中大型 Web 屬性提供可擴展性和可管理性功能。

### Important

您可以使用 SQL Server Web Edition 僅支援公有和可存取網際網路的網頁、網站、Web 應用程式和 Web 服務。您無法使用 SQL Server Web Edition line-of-business (例如客戶關係管理或企業資源管理應用程式)。

- 開發人員 – SQL Server Developer Edition 包含 Enterprise Edition 的所有功能，但僅用於開發目的。
- Express – SQL Server Express Edition 是免費的資料庫，可用於學習或建置桌面應用程式。您可以將 Express Edition 更新為其他版本。

### Note

SQL Server 評估版本提供 180 天的試用期。

## 成本影響

您可以從 Microsoft 經銷商購買 SQL Server 授權，並使用 AWS 軟體保證將授權帶到。或者，您可以使用 SQL Server 授權搭配 pay-as-you-go 模型。Amazon EC2 AMIs

如果您從 Microsoft 經銷商購買 SQL Server 授權，核心授權會以兩個套件的形式販售，而且每部伺服器必須至少授權四個核心。下表顯示 Enterprise 版和 Standard 版之間的成本比較。

| 版本   | SQL Server Enterprise Edition (2 個核心套件) | SQL Server Standard Edition (2 個核心套件) | 節省  |
|------|---|---------------------------------------|-----|
| 2022 | 15,123 美元                               | 3,945 美元                              | 74% |
| 2019 | 13,748 美元                               | 3,586 美元                              | 74% |

**Note**

上表中的定價是以 Microsoft 的 [SQL Server 2022](#) 和 [SQL Server 2019](#) 公開定價為基礎。

下列成本比較顯示使用包含授權的 Amazon EC2 AMIs 託管不同版本的 SQL Server。在此比較中，SQL Server 託管在 us-east-1 區域的 r6i.xlarge (4 個 vCPU) 上。

| 執行個體                                 | 運算成本      | Windows 授權成本 | SQL Server 授權成本 | 總計          |
|--------------------------------------|-----------|--------------|-----------------|-------------|
| R6i.xlarge (Linux)                   | 183.96 美元 | –            | –               | 183.96 美元   |
| R6i.xlarge + Windows                 | 183.96 美元 | 134.32 美元    | –               | 318.28 美元   |
| R6i.xlarge + SQL Server Web 版本       | 183.96 美元 | 134.32 美元    | 49.35 美元        | 367.63 美元   |
| R6i.xlarge + SQL Server Standard 版   | 183.96 美元 | 134.32 美元    | 350.4 美元        | 668.68 美元   |
| R6i.xlarge + SQL Server Enterprise 版 | 183.96 美元 | 134.32 美元    | 1,095 美元        | 1,413.28 美元 |

您可以為您的工作負載選取正確的 SQL Server 版本，以節省高達 95% 的 SQL Server 授權成本。下表比較 r6i.xlarge 執行個體上的 SQL Server 授權成本。

| 版本                   | 節省 % |
|----------------------|------|
| 標準與企業的比較             | 68%  |
| Web 與標準相比            | 86%  |
| Web 與 Enterprise 的比較 | 95%  |

在大多數情況下，組織會從 Enterprise 切換到 Standard Edition，但在某些情況下，可以從 Standard 或 Enterprise Edition 切換到 Web Edition。

## 成本最佳化建議

您可以根據擴展限制、高可用性、效能和安全性，為您的工作負載選擇最佳版本。下表顯示 SQL Server 版本支援的功能。這可協助您決定要使用哪個版本。此比較適用於 [SQL Server 2016 SP1 和更新版本](#)。

### 擴展限制

下表比較不同 SQL Server 版本的擴展限制。

| 功能  | 企業版    | 標準版本                 | Web 版本               | Express 版本          |
|---|--------|----------------------|----------------------|---------------------|
| SQL Server Database Engine、SQL Server Analysis Services (SSAS) 或 SQL Server Reporting Services (SSRS) 的單一執行個體所使用的最大運算容量 | 作業系統上限 | 限制為小於 4 個通訊端或 24 個核心 | 限制為少於 4 個通訊端或 16 個核心 | 限制為小於 4 個通訊端或 4 個核心 |
| SQL Server Database Engine 每個執行個體緩  | 作業系統上限 | 128 GB               | 64 GB                | 1410 MB             |

| 功能                                | 企業版           | 標準版本         | Web 版本 | Express 版本 |
|-----------------------------------|---------------|--------------|--------|------------|
| 衝集區的記憶體上限                         |               |              |        |            |
| SQL Server 資料庫引擎每個執行個體的緩衝集區延伸容量上限 | 已設定記憶體上限 32 倍 | 設定的記憶體上限 4 倍 | N/A    | N/A        |
| 關聯式資料庫大小上限                        | 524 PB        | 524 PB       | 524 PB | 10 GB      |
| 資料欄存放區快取或記憶體最佳化資料的記憶體上限           | 作業系統上限        | 32 GB        | 16 GB  | 352 MB     |

如果您的應用程式需要少於 16 個核心 (32 個 vCPUs) 和 64 GB 的 RAM，則您可以從 SQL Server Web 版本開始評估。如果您的工作負載需要超過 64 GB 的記憶體或其他高可用性選項，則必須升級至 SQL Server Standard Edition。

您可以使用 SQL Server Web Edition 來支援公有和網際網路可存取的網頁、網站、Web 應用程式和 Web 服務，但無法使用 SQL Server Web Edition 來支援業務應用程式。如需 SQL Server Web Edition 使用案例的詳細資訊，請聯絡 [Microsoft 授權支援](#) 或您的 Microsoft 經銷商。

針對最多 24 個核心 (48 個 vCPUs) 和 128 GB 記憶體的工作負載，您可以使用 SQL Server Standard Edition。不過，您可以使用 [緩衝集區延伸](#)，讓 SQL Server Standard Edition 能夠利用 [本機執行個體儲存體](#)，例如 r6id EC2 執行個體中存在的執行個體。這可將記憶體大小擴充至最大記憶體組態的四倍。當記憶體需求開始增加時，這種功能組合可能會延遲伺服器必須升級至 Enterprise Edition。

您可以透過尋找緩衝集區中的資料庫頁面和 [頁面預期壽命](#) 計數器來識別記憶體使用率。頁面預期壽命會告訴您頁面在排回磁碟之前在記憶體中的時間。此計數器預設值為 300。如果頁面位於記憶體中數小時或數天，則有機會減少配置的記憶體。

## 高可用性

下表比較不同 SQL Server 版本的高可用性功能。

| 功能                           | 企業版                             | 標準版本   | Web 版本 | Express 版本 |
|------------------------------|---------------------------------|--------|--------|------------|
| 伺服器核心支援<br>1                 | 是                               | 是      | 是      | 是          |
| 日誌傳送                         | 是                               | 是      | 是      | 否          |
| 資料庫鏡像                        | 是                               | 完整安全模式 | 僅做為見證  | 僅做為見證      |
| 備份壓縮                         | 是                               | 是      | 否      | 否          |
| Always On 容錯<br>移轉叢集執行個<br>體 | 16 個節點                          | 2 個節點  | 否      | 否          |
| Always On 可用<br>性群組          | 最多 8 個次要複<br>本，包括 2 個同<br>步次要複本 | 否      | 否      | 否          |
| 基本可用性群組                      | 否                               | 2 個節點  | 否      | 否          |
| 線上頁面和檔案<br>還原                | 是                               | 否      | 否      | 否          |
| 線上索引                         | 是                               | 否      | 否      | 否          |
| 線上結構描述變<br>更                 | 是                               | 否      | 否      | 否          |
| 快速復原                         | 是                               | 否      | 否      | 否          |
| 鏡像備份                         | 是                               | 否      | 否      | 否          |
| 熱新增記憶體和<br>CPU               | 是                               | 否      | 否      | 否          |
| 加密備份                         | 是                               | 是      | 否      | 否          |

| 功能                              | 企業版 | 標準版本 | Web 版本 | Express 版本 |
|---------------------------------|-----|------|--------|------------|
| 混合備份到 Microsoft Azure (備份到 URL) | 是   | 是    | 否      | 否          |
| 用於災難復原的容錯移轉伺服器                  | 是   | 是    | 否      | 否          |
| 提供高可用性的容錯移轉伺服器                  | 是   | 是    | 否      | 否          |

## 其他常見功能

下表比較不同 SQL Server 版本的最常見功能。如需廣泛的功能清單，請參閱 Microsoft 文件中的 [SQL Server 2019 版本和支援的功能](#)。

| 功能                  | 企業版 | 標準版本 | Web 版本 | Express 版本 |
|---------------------|-----|------|--------|------------|
| (效能) 資源總監           | 是   | 否    | 否      | 否          |
| (安全性) 透明資料庫加密 (TDE) | 是   | 是    | 否      | 否          |
| (安全性) 可擴展金鑰管理 (EKM) | 是   | 否    | 否      | 否          |
| (複寫) Oracle 發佈      | 是   | 否    | 否      | 否          |
| (複寫) 點對點交易複寫        | 是   | 否    | 否      | 否          |
| 變更資料擷取              | 是   | 是    | 否      | 否          |

## SQL Server 開發人員版本

所有非生產工作負載，例如開發、QA、測試、預備和 UAT 環境，都可以使用 SQL Server Developer Edition 節省 100% 的 SQL Server 授權成本。[下載 SQL Server](#) 之後，您可以使用共用租用在 EC2 執行個體上安裝 SQL Server Developer Edition。SQL Server Developer Edition 不需要專用基礎設施。如需詳細資訊，請參閱本指南的 [SQL Server 開發人員版本](#) 建議。

## 切換版本

對於現有的工作負載，從一個版本切換到另一個版本需要廣泛的測試。最佳實務是檢查在 Enterprise 或 Standard Edition 上執行的工作負載，以查看是否使用版本特定的功能，以及這些功能是否有任何替代解決方案。例如，如果您想要查看資料庫是否使用任何企業級功能，您可以執行動態[管理檢視 \(DMV\)](#)，如下列範例命令所示。

```
SELECT feature_name FROM sys.dm_db_persisted_sku_features; GO
```

有些 Enterprise Edition 功能無法在 T-SQL 中擷取，例如線上重新索引做為 SQL 維護任務的一部分。這些必須手動驗證。

## 遷移考量事項

您授權 SQL Server 的方式將決定切換版本的選項。AMIs，包括 SQL Server AMIs，其授權成本包含在 EC2 執行個體的價格中，授權成本受限於 AMI。您可以使用[AWS 帳單代碼](#)來驗證 AMI 中包含的 SQL Server 版本。對於包含 AWS 授權的執行個體，在作業系統內變更 SQL Server 版本不會變更與 AMI 相關聯的帳單。您必須使用執行新版 SQL Server 的 AMI，將資料庫遷移至新的 EC2 執行個體。

如果您使用自己的授權，則有更多彈性。通常仍建議遷移至執行新版本的另一個 EC2 執行個體。如果某件事未按計劃進行，這可讓容錯回復更為簡單。不過，如果您必須使用現有的伺服器，您仍然可以 side-by-side 安裝 SQL Server，並在執行個體之間遷移資料庫。如需 side-by-side 版本降級的詳細資訊，請參閱 MSSQLTips 網站上的 [SQL Server 中的版本升級和降級](#)。

## 其他資源

- [SQL Server 2022 的版本和支援的功能](#) (Microsoft Learn)
- [sys.dm\\_db\\_persisted\\_sku\\_features \(Transact-SQL\)](#) (Microsoft Learn)
- [您應該使用哪個版本的 SQL Server ?](#) (Brent Ozar Unlimited)
- [AWS 定價計算工具](#) (AWS)

# 評估 SQL Server 開發人員版本

## 概觀

[SQL Server Developer Edition](#) 是 SQL Server 的免費版本，其中包含 Enterprise Edition 的所有功能，可用於任何非生產環境。在無法使用 Microsoft Developer Network (MSDN) 授權的雲端中，SQL Server Developer Edition 是節省成本的好方法，而無需提供開發和測試工作負載的授權。對於執行大型開發和測試環境，並尋求降低不必要成本的團隊來說尤其如此。

生產環境定義為由應用程式最終使用者（例如網際網路網站）存取的環境，其用途不只是收集該應用程式的意見回饋或接受測試。構成生產環境的其他案例包括：

- 連線至生產資料庫的環境
- 支援生產環境災難復原或備份的環境
- 至少在某些時間用於生產的環境，例如在活動尖峰期間輪換至生產環境的伺服器

如需授權的詳細資訊，請參閱 文件中的 AWS [Amazon Web Services](#) 和 Microsoft：[常見問答集](#)。

## 成本影響

如果您將 SQL Server Developer Edition 用於非生產工作負載，則可以為開發和測試環境節省 100% 的目前 SQL Server 授權成本。

| SQL Server 版本 | SQL Server Enterprise Edition (2 個核心套件) | SQL Server Standard 版 (2 個核心套件) | SQL Server 開發人員版本 |
|---------------|---|---------------------------------|-------------------|
| 2022          | 15,123 美元                               | 3,945 美元                        | 免費                |
| 2019          | 13,748 美元                               | 3,586 美元                        | 免費                |

### Note

上表中的定價是以 Microsoft 的 [SQL Server 2022](#) 和 [SQL Server 2019](#) 公開定價為基礎。

下表比較使用 4 個 vCPUs 和 us-east-2 區域中隨需定價執行的不同 SQL Server 版本的成本。這適用於依賴包含授權之執行個體的案例 AWS。

| EC2 執行個體  | AMI  | 運算價格      | Windows 授權價格 | SQL Server 授權價格 | 總價         |
|-----------|--|-----------|--------------|-----------------|------------|
| r5.xlarge | Linux ( 運算定價 )                               | 183.96 美元 | –            | –               | 183.96 美元  |
| r5.xlarge | Linux + SQL Server 開發人員版本                    | 183.96 美元 | 0 USD        | 0 USD           | 183.96 美元  |
| r5.xlarge | Windows Server (LI)                          | 183.96 美元 | 134.32 美元    | –               | 318.28 美元  |
| r5.xlarge | Windows + SQL Server 開發人員版本                  | 183.96 美元 | 134.32 美元    | 0 USD           | 318.28 美元  |
| r5.xlarge | Windows + SQL Server Web 版本 (LI)             | 183.96 美元 | 134.32 美元    | 49.64 美元        | 367.92 美元  |
| r5.xlarge | Windows + SQL Server Standard Edition (LI)   | 183.96 美元 | 134.32 美元    | 350.4 美元        | 668.68 美元  |
| r5.xlarge | Windows + SQL Server Enterprise Edition (LI) | 183.96 美元 | 134.32 美元    | 1095 美元         | 1413.28 美元 |

## 成本最佳化案例

資料完整性公司進行新的取得後，想要將新取得的工作負載從受管託管供應商的目前位置遷移，以與中的其他工作負載合併 AWS 雲端。初始定價顯示，與目前的受管服務供應商 AWS 相比，公司的 SQL

Server 工作負載在上執行的成本高出 60%。MACO SME 評估了預估值，發現客戶實際在受管託管供應商為開發和測試環境支付 SQL Server 授權的費用。透過在遷移期間將非生產工作負載切換到 SQL Server 開發人員版本，公司減少了 40% 的 SQL Server 授權。

## Amazon EC2 上包含的 SQL Server 授權

如果您的 EC2 執行個體上有使用[包含授權的 AMIs](#) SQL Server，則無法從 Enterprise Edition 直接轉換為 Developer Edition。包含授權之執行個體的授權成本受限於 AMI。即使 SQL Server 從作業系統內解除安裝，EC2 執行個體仍需支付授權成本。

若要轉換為開發人員版本，您必須[下載 SQL Server 開發人員版本](#)，將其安裝在新的 EC2 執行個體上，然後遷移資料庫。您可以使用各種方法在 EC2 執行個體之間遷移 SQL Server 資料庫。如需詳細資訊，請參閱將 Microsoft AWS 雲端 [SQL Server 資料庫遷移至指南中的 SQL Server 資料庫遷移方法](#)。您也可以使用[自動化 SQL Server Developer 解決方案](#)來準備您計劃遷移到的新執行個體。

## Amazon EC2 上的 SQL Server BYOL

如果您有使用 BYOL 的 SQL Server 執行個體，您可以選擇下列就地轉換或side-by-side降級選項：

- 從 Microsoft 網站下載 [SQL Server 開發人員版本](#)。如需手動或自動安裝說明，請參閱 AWS 部落格上的[自動化 SQL Server Developer 部署](#)文章。
- 使用 [SQL Server 原生備份和還原](#)來遷移資料庫，或從一個 SQL 執行個體分離/連接資料庫到另一個執行個體。
- 使用 [自動化工具](#)進行大量部署。

### Note

SQL Server Developer Edition 僅適用於非生產環境。

## 其他資源

- [自動化 SQL Server Developer 部署以在 EC2 上部署 SQL Server Developer Edition](#) (AWS 部落格)
- [SQL 2022 定價](#) (Microsoft)
- [SQL 2019 定價](#) (Microsoft)
- [授權選項](#) (Amazon EC2 上的 SQL Server)

- [AWS 定價計算工具](#) (Amazon EC2 上的 SQL Server 文件 )
- [Microsoft SQL Server 2019 授權指南](#) ( 從 Microsoft 下載 )
- [SQL Server 2022 開發人員版本](#) ( 從 Microsoft 下載 )

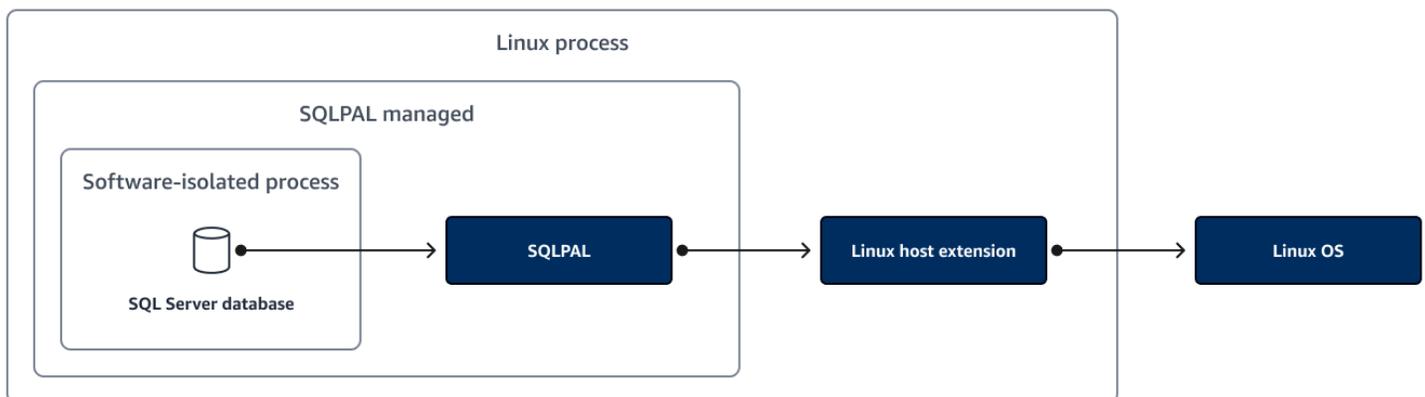
## 評估 Linux 上的 SQL Server

### 概觀

自 SQL Server 2017 起，可以在 Linux 作業系統上安裝 SQL Server。Linux 上的 SQL Server 已就緒，可提供彈性、高效能、安全功能、降低 TCO、HA/DR 功能，以及絕佳的使用者體驗。您可以從 Windows Server 上的 SQL Server 切換至 Linux 上的 SQL Server，以節省 Windows Server 授權成本。

對於 Linux，SQL Server 可在 Red Hat Enterprise Linux (RHEL)、SUSE Linux Enterprise Server (SLES)、Ubuntu 和 Amazon Linux 2 上部署。SQL Server 資料庫引擎在 Windows Server 和 Linux 上執行的方式相同，但在使用 Linux 時，某些任務有一些基本變更。在 Linux 和 Windows 上執行 SQL Server Always On 應用程式的一個關鍵區別是容錯移轉叢集。如果您在 Windows Server 主機上部署 Always On 可用性群組，則可以利用 [Windows Server 容錯移轉叢集 \(WSFC\)](#) 和 Active Directory 作為支援容錯移轉叢集的內建功能。不過，WSFC 和 Active Directory 都不支援 Linux 上的容錯移轉叢集。如果您想要在 Linux 上啟動 SQL Server 的容錯移轉叢集，您可以使用 [ClusterLabs Pacemaker AWS Launch Wizard](#) 來簡化 Linux 執行個體上的叢集設定和 SQL 安裝。

Windows 和 Linux 上的 SQL Server 共用共同的程式碼庫。也就是說，SQL Server 核心引擎完全未變更為在 Linux 上執行。SQL Server 推出平台抽象層 (SQLPAL)，如下圖所示。



SQLPAL 負責抽象化 SQL Server 與基礎作業系統之間的呼叫和通訊。主機擴充功能只是原生 Linux 應用程式。低階作業系統函數是原生呼叫，用於最佳化 I/O、記憶體和 CPU 用量。當主機延伸項目啟動

時，它會載入並初始化 SQLPAL，然後啟動 SQL Server。SQLPAL 會啟動隔離的軟體程序，以提供其餘程式碼所需的翻譯。將此新層新增至 SQL Server 架構表示無論作業系統為何，都可以使用讓 SQL Server 在 Windows 上如此強大的相同企業級核心功能和優點。

## 成本影響

對於 r5.2xlarge 執行個體，每個案例中的 Windows Server 授權成本降低約為 268 美元。與使用價格較低的 SQL Server 版本相比，減少是伺服器總成本的較高百分比。下表顯示節省的成本。

| 執行個體       | 版本         | Windows 上的 SQL Server 每月成本 | Linux 上的 SQL Server 每月成本 | 節省  |
|------------|------------|----------------------------|--------------------------|-----|
| r5.2xlarge | Web        | 735 美元                     | 466 美元                   | 37% |
| r5.2xlarge | 標準         | 1,337 美元                   | 1,068 美元                 | 20% |
| r5.2xlarge | Enterprise | 2,826 美元                   | 2,558 美元                 | 10% |

### Note

上表中的價格估算是根據 us-east-1 區域中的隨需定價，並且可以直接在 [中檢視 AWS 定價計算工具](#)。

考慮一個範例案例，其中 SMB 客群中的 ISV 客戶希望為其開發環境節省成本。他們已在一組 Windows 伺服器上使用 SQL Server Developer Edition。透過從 Windows with SQL Server Developer Edition 切換到 Linux with SQL Server Developer Edition，ISV 客戶可以在其開發工作負載上節省 33%。下表顯示此案例的下列預估成本。

| 估算                                   | 每月成本              |
|--------------------------------------|-------------------|
| <a href="#">Windows + SQL Server</a> | 9,307.72 美元       |
| <a href="#">Linux + SQL Server</a>   | 6,218.36 美元       |
| 預估節省成本                               | 3,089.36 美元 (33%) |

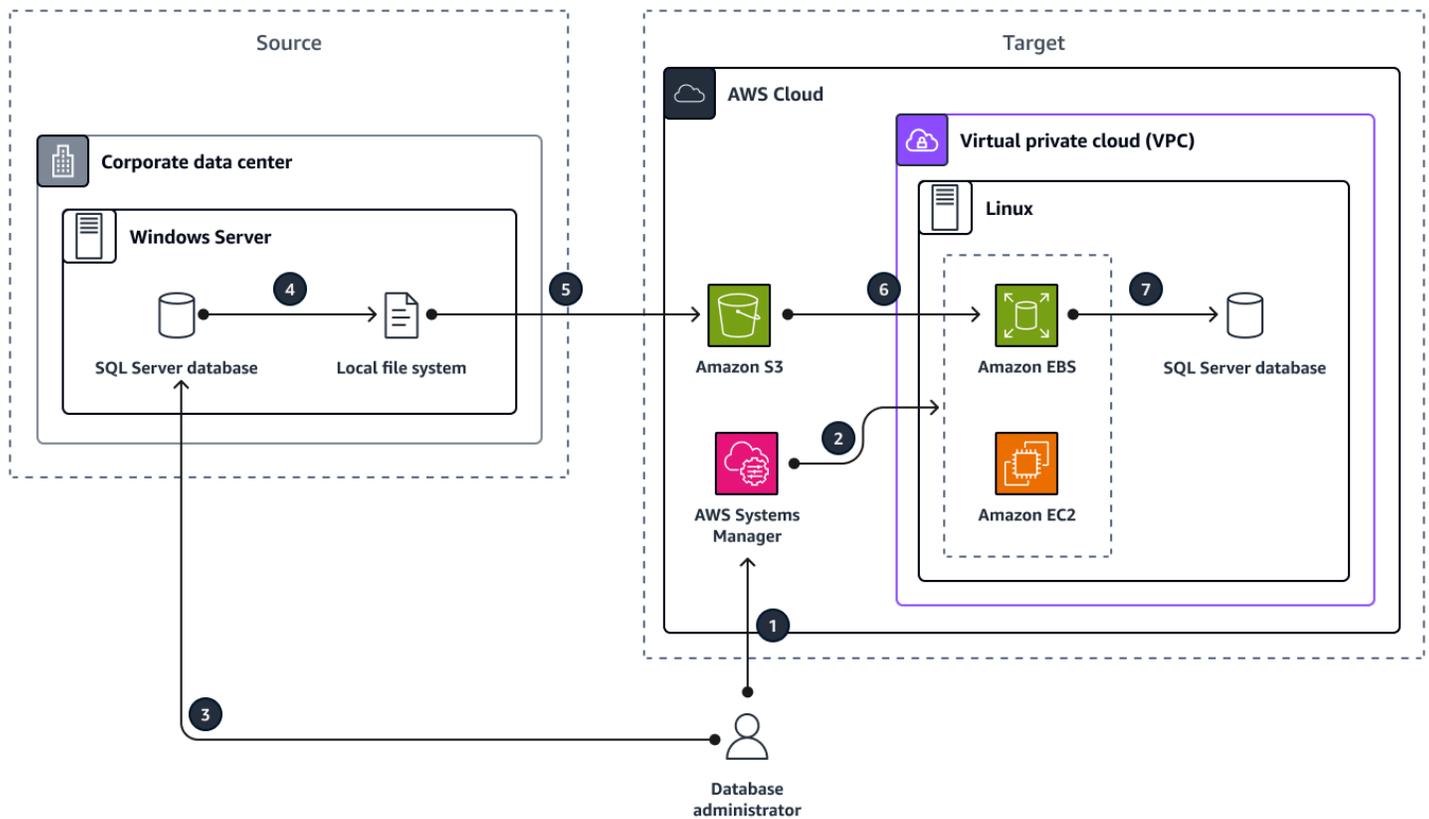
在另一個範例案例中，公司會將包含授權的 SQL Server EC2 執行個體從 Windows 遷移至 Linux。該公司每年節省總計 300,000 USD 的 Windows Server 授權成本，大約是總 AWS 帳單的 20%。

## 成本最佳化建議

我們建議您考慮下列事項：

- 從 SQL Server 2017 開始支援 Linux 上的 SQL Server。
- 若要協助進行切換，您可以使用[適用於 Microsoft SQL Server 資料庫的 Windows 到 Linux 轉換助理](#)。重建助理是一種指令碼工具，可協助您將現有的 SQL Server 工作負載從 Windows 移至 Linux 作業系統，方法是檢查是否有常見的不相容、從 Windows 主機匯出資料庫，然後將資料庫匯入在 Ubuntu 16.04 上執行 Microsoft SQL Server 2017 的 EC2 執行個體。
- 您也可以使用 SQL Server 中的[備份和還原](#)功能，從 Windows 上的 SQL Server 切換到 Linux。
- 您可以使用輕鬆快速地部署到 Linux 或 Ubuntu 上的 SQL Server [AWS Launch Wizard](#)。啟動精靈可以根據您的應用程式需求，在獨立和高可用性案例中部署 Linux 或 Ubuntu 上的 SQL Server。如需詳細資訊，請參閱 AWS 部落格 Microsoft 工作負載中的[部署至 SQL Server Always on Linux AWS Launch Wizard](#) 中的文章。

下圖顯示 解決方案的架構，該解決方案使用適用於 Microsoft SQL Server 資料庫的 Windows 到 Linux 轉換助理。



## 其他資源

- [Linux 上的 SQL Server 概觀](#) (Microsoft Learn)
- [Linux 上的 SQL Server 安裝指南](#) (Microsoft Learn)
- [使用 部署至 SQL Server Always on Linux AWS Launch Wizard](#) ( AWS 部落格上的 Microsoft 工作負載 )
- [Linux 上的高可用性 SQL Server](#) (AWS 開放原始碼部落格 )

## 最佳化 SQL Server 備份策略

### 概觀

大多數組織正在尋找正確的解決方案來保護其在 [Amazon EC2](#) 上的 SQL Server 上的資料，以滿足其目前對復原點目標 (RPO)、自上次備份以來可接受的時間上限，以及復原時間目標 (RTO)、服務中斷和服務還原之間的可接受延遲上限的需求。如果您在 EC2 執行個體上執行 SQL Server，您有多個選項可建立資料的備份並還原資料。保護 Amazon EC2 上 SQL Server 資料的備份策略包括：

- 使用已啟用 Windows 磁碟區陰影複製服務 (VSS) 的 [Amazon Elastic Block Store \(Amazon EBS\)](#) 快照或 [AWS Backup](#) 的伺服器層級備份
- 在 SQL Server 中使用 [原生備份和還原的資料庫層級備份](#)

您具有下列[資料庫層級原生備份](#)的儲存選項：

- 具有 [Amazon EBS 磁碟區](#)的本機備份
- 使用 [Amazon FSx for Windows File Server](#) 或 [Amazon FSx for NetApp ONTAP](#) 進行網路檔案系統備份 FSx NetApp
- 使用 將網路備份至 Amazon Simple Storage Service (Amazon S3) [AWS Storage Gateway](#)
- 直接備份至 Amazon S3 for SQL Server 2022

本節執行下列動作：

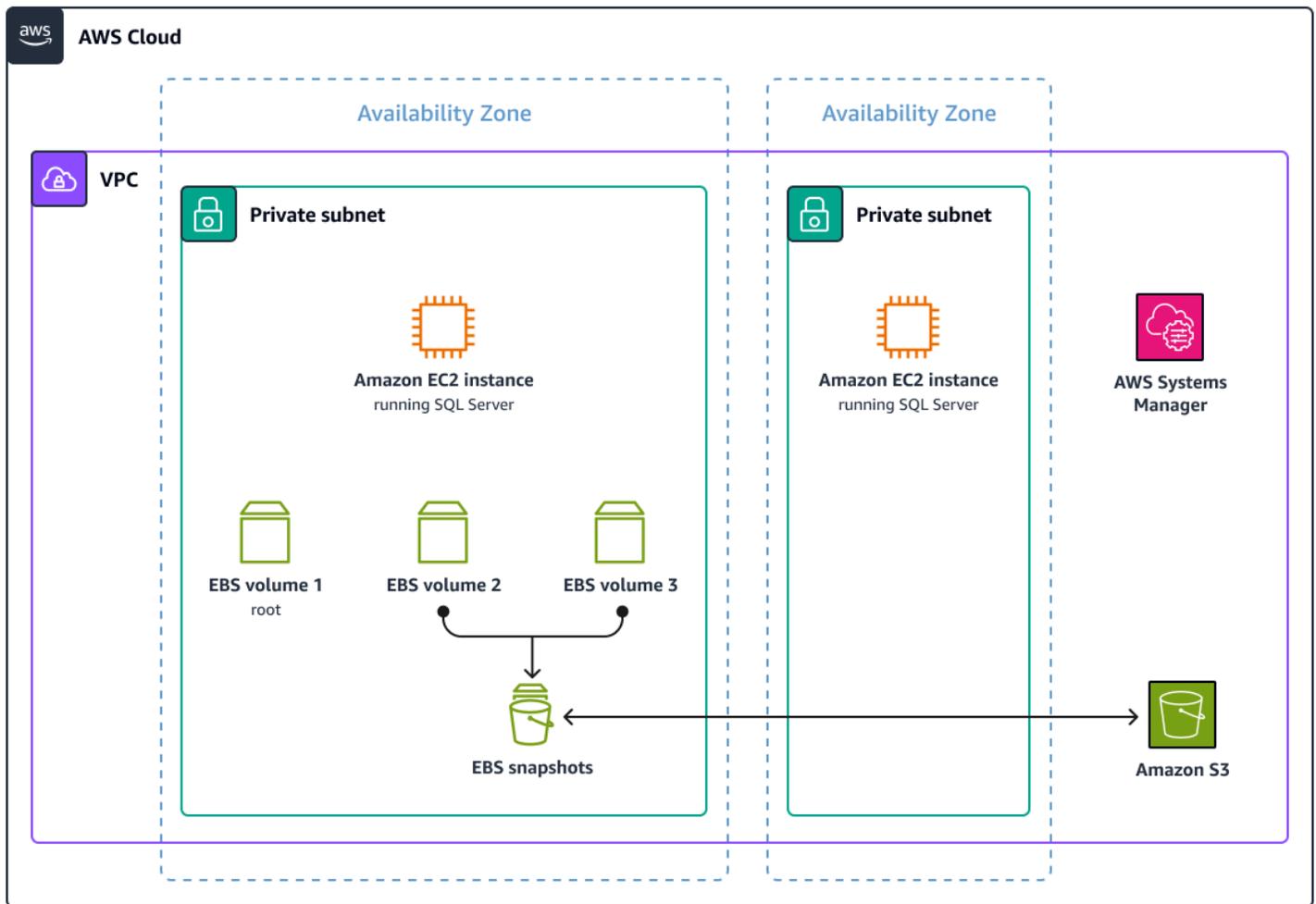
- 反白顯示可協助您節省儲存空間的功能
- 比較不同後端儲存選項之間的成本
- 提供深入文件的連結，以協助實作這些建議

## 使用啟用 VSS 的快照進行伺服器層級備份

啟用 VSS 的快照架構會使用 AWS Systems Manager [執行命令](#)，在您的 SQL Server 執行個體上安裝 VSS 代理程式。也可以使用執行命令，調用將作業系統和應用程式緩衝區清空至磁碟、暫停 I/O 操作、獲取 EBS 磁碟區的時間點快照、然後恢復 I/O 的整個工作流程。

此執行命令會建立附接至目標執行個體之所有 EBS 磁碟區的自動快照。也可以選擇排除根磁碟區，因為使用者資料庫檔案通常儲存在其他磁碟區上。如果分割多個 EBS 磁碟區，以便為 SQL Server 檔案建立單一檔案系統，Amazon EBS 也支援使用單一 API 命令建立當機一致的多磁碟區快照。如需啟用應用程式一致 [VSS 的 EBS 快照](#)的詳細資訊，請參閱 Amazon EC2 文件中的[建立 VSS 應用程式一致快照](#)。

下圖顯示使用已啟用 VSS 的快照進行伺服器層級備份的架構。



請考慮下列使用具備 VSS 功能的快照的優點：

- 資料庫執行個體的第一個快照會包含完整的資料庫執行個體資料。相同資料庫執行個體的后續快照均採用**遞增**處理，這表示系統僅會儲存上次執行快照後有所變更的資料。
- EBS 快照提供point-in-time復原。
- 可以**從快照還原到新的 SQL Server EC2 執行個體**。
- 如果執行個體是使用 Amazon EBS 加密，或者如果資料庫是使用 TDE 在執行個體中加密，則該執行個體或資料庫會使用相同的加密自動還原。
- 您可以複製**跨區域的自動備份**。
- 當您從快照還原 EBS 磁碟區時，應用程式可立即存取該磁碟區。這表示您可以在從快照還原一個或多個基礎 EBS 磁碟區之後，立即讓 SQL Server 上線。
- 依預設，還原的磁碟區會在應用程式第一次嘗試讀取時，從 Amazon S3 中擷取基礎區塊。這表示從快照還原 EBS 磁碟區後，效能可能會存在延遲。音量最終達到標稱性能。但是，可以透過使用**快速快照還原 (FSR)** 快照來避免這種延遲。

- 可以使用 [EBS 快照的生命週期管理](#)。

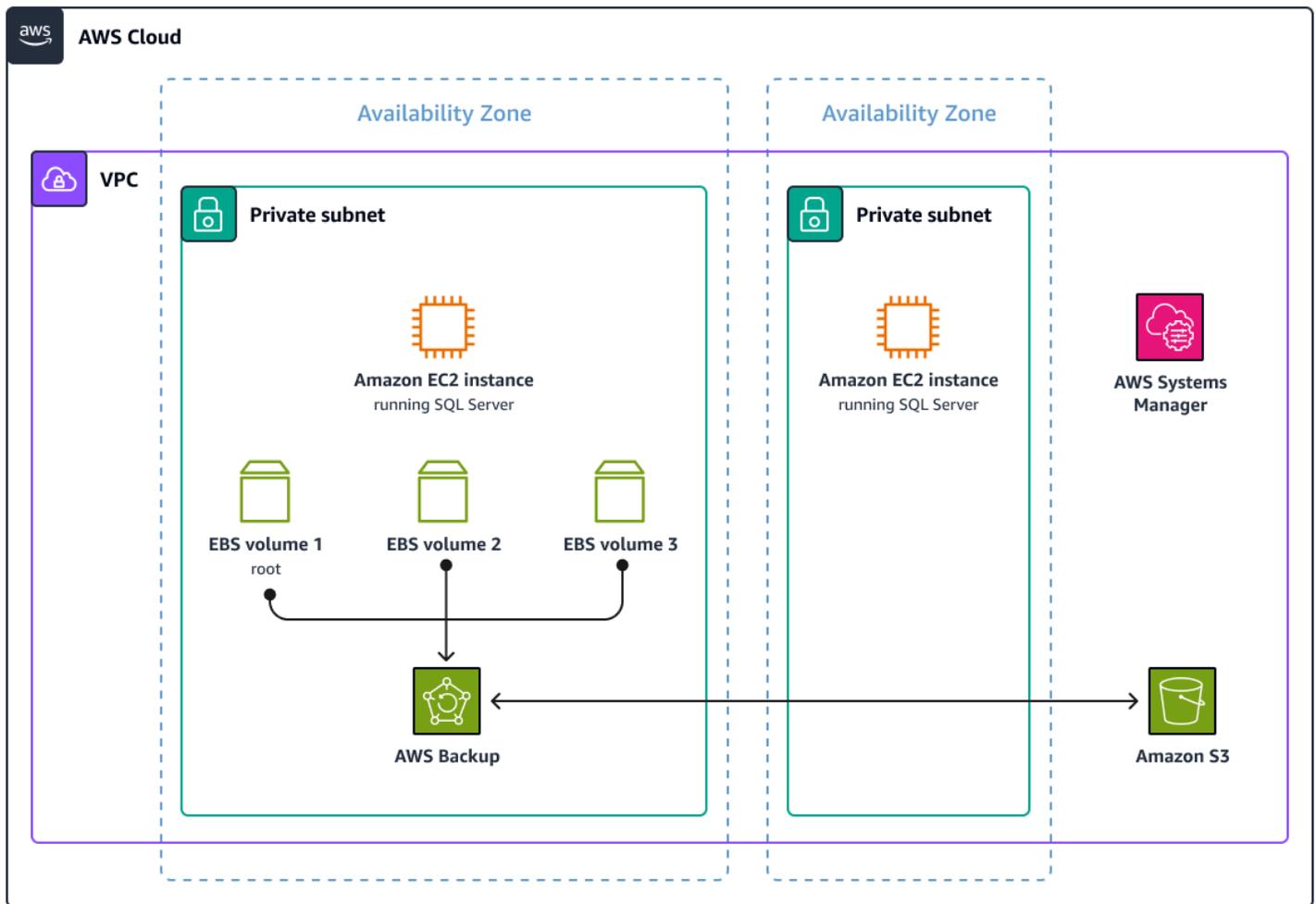
請考慮下列使用已啟用 VSS 快照的限制：

- 您無法使用 SQL Server 執行個體的加密快照執行跨區域point-in-time復原。
- 您無法建立未加密執行個體的加密快照。
- 您無法還原個別資料庫，因為快照是在 EBS 磁碟區層級拍攝。
- 您無法將執行個體還原至 本身。
- 資料庫執行個體的快照必須使用與資料庫執行個體相同的 AWS Key Management Service (AWS KMS) 金鑰加密。
- 在快照備份過程中，儲存 I/O 會暫停幾秒鐘 (約 10 毫秒)。

## 使用的 SQL Server 備份 AWS Backup

您可以使用來[AWS Backup](#)集中和自動化跨 的資料保護 AWS 服務。AWS Backup 提供具成本效益、全受管、以政策為基礎的解決方案，可簡化大規模的資料保護。AWS Backup 也可協助您支援法規合規義務，並實現業務持續性目標。與一起 AWS Organizations，AWS Backup 可讓您集中部署資料保護（備份）政策，以設定、管理和管理整個組織 AWS 帳戶 和資源的備份活動。

下圖顯示 EC2 上 SQL Server 備份和還原解決方案的架構 AWS Backup。



請考慮使用 備份 SQL Server 的下列優點 AWS Backup：

- 可自動化備份排程、保留管理和生命週期管理。
- 您可以集中整個組織的備份策略，橫跨多個帳戶和 AWS 區域。
- 您可以集中監控備份活動，並跨 發出提醒 AWS 服務。
- 可以為災難復原規劃實作跨區域備份。
- 該解決方案可支援跨帳戶備份。
- 可以使用次要備份加密執行安全備份。
- 所有備份都支援使用加密金鑰進行 AWS KMS 加密。
- 該解決方案可於 TDE 搭配使用。
- 可以從 AWS Backup 主控台還原至特定復原點。
- 可以備份整個 SQL Server 執行個體，其中包括所有 SQL Server 資料庫。

## 資料庫層級備份

這些方法使用原生 Microsoft SQL Server 備份功能。您可以對 SQL Server 執行個體上的個別資料庫進行備份，並還原個別資料庫。

原生 SQL Server 備份和還原的每個選項還支援下列各項：

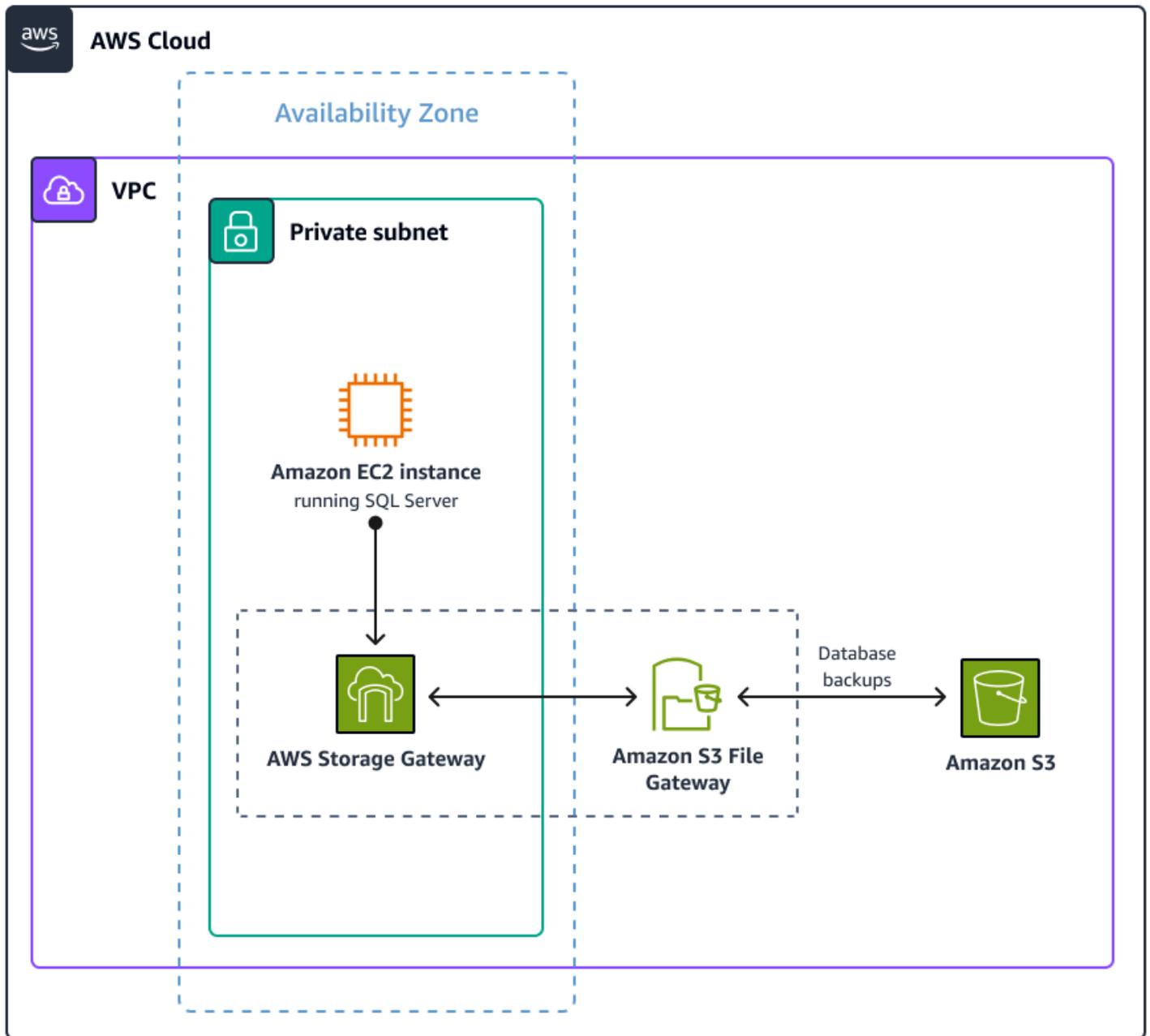
- 壓縮和多檔案備份
- 完整、差異和 T 日誌備份
- TDE 加密的資料庫

### SQL Server 原生備份和還原至 Amazon S3

Amazon EC2 上的 SQL Server 支援 SQL Server 資料庫的原生備份和還原。您可以對 SQL Server 資料庫進行備份，然後將備份檔案還原到現有的資料庫或新的 SQL Server EC2 執行個體、Amazon RDS for SQL Server 或內部部署伺服器。

Storage Gateway 是一種混合雲端儲存服務，可讓內部部署應用程式存取幾乎無限制的雲端儲存。您可以使用 Storage Gateway 將 Microsoft SQL Server 資料庫直接備份到 Amazon S3，減少內部部署儲存體使用量，並使用 Amazon S3 提供耐用、可擴展且符合成本效益的儲存體。

下圖顯示使用 Storage Gateway 和 Amazon S3 的原生備份和還原解決方案的架構。



考慮將原生 SQL Server 備份與 Storage Gateway 搭配使用的下列優點：

- 您可以將儲存閘道映射為 EC2 執行個體上的 Server Message Block (SMB) 檔案共用，然後將備份傳送至 Amazon S3。
- 備份會直接前往 S3 儲存貯體或透過 Storage Gateway 檔案快取。
- 支援多檔案備份。

考慮使用 Storage Gateway 進行原生備份的下列限制：

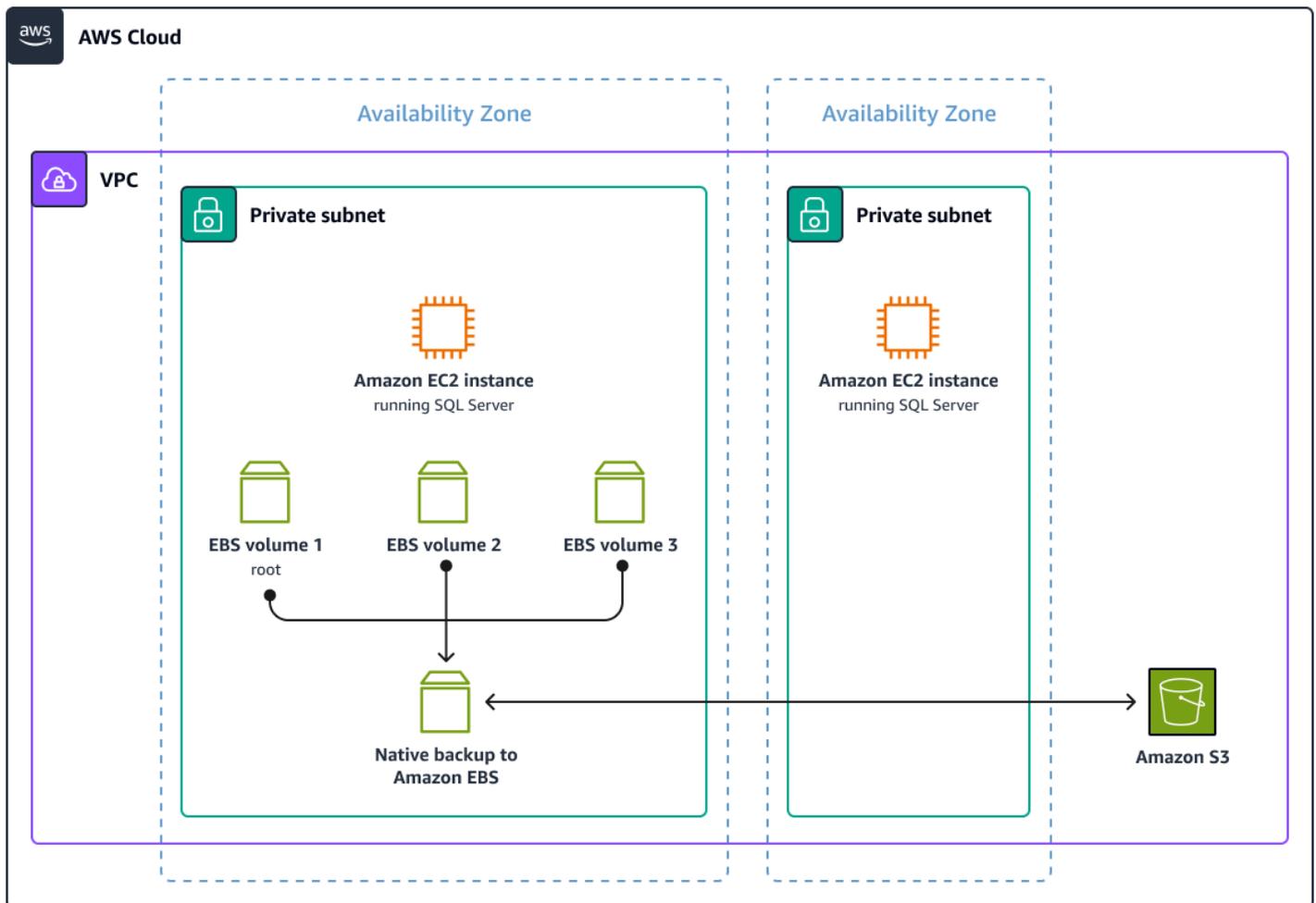
- 必須為每個單獨的資料庫設定備份和還原。
- 必須管理備份檔案的 [Amazon S3 生命週期政策](#)。

如需如何設定 Storage Gateway 的詳細資訊，請參閱 AWS 部落格上的[使用文章在 Amazon S3 中存放 SQL Server 備份 AWS Storage Gateway](#)。

## SQL Server 原生備份至 EBS 磁碟區

您可以對 SQL Server 資料庫進行原生備份，並將檔案存放在 Amazon EBS 磁碟區中。Amazon EBS 是高效能的區塊儲存服務。EBS 磁碟區是彈性的，支援加密。它們可以分離並連接到 EC2 執行個體。您可以在相同 EBS 磁碟區類型或不同 EBS 磁碟區類型上的 EC2 執行個體上備份 SQL Server。備份至不同 EBS 磁碟區的優點之一是節省成本。

下圖顯示原生備份至 EBS 磁碟區的架構。



請考慮使用 SQL Server 原生備份到 EBS 磁碟區的下列優點：

- 您可以備份 SQL Server EC2 執行個體上的單獨資料庫，並還原單獨資料庫，而不必還原整個執行個體。
- 支援多檔案備份。
- 可以使用 SQL Server Agent 和 SQL Server 作業引擎來排程備份作業。
- 可以透過硬體選擇獲得效能優勢。例如，可以使用 st1 儲存磁碟區來達到更高的輸送量。

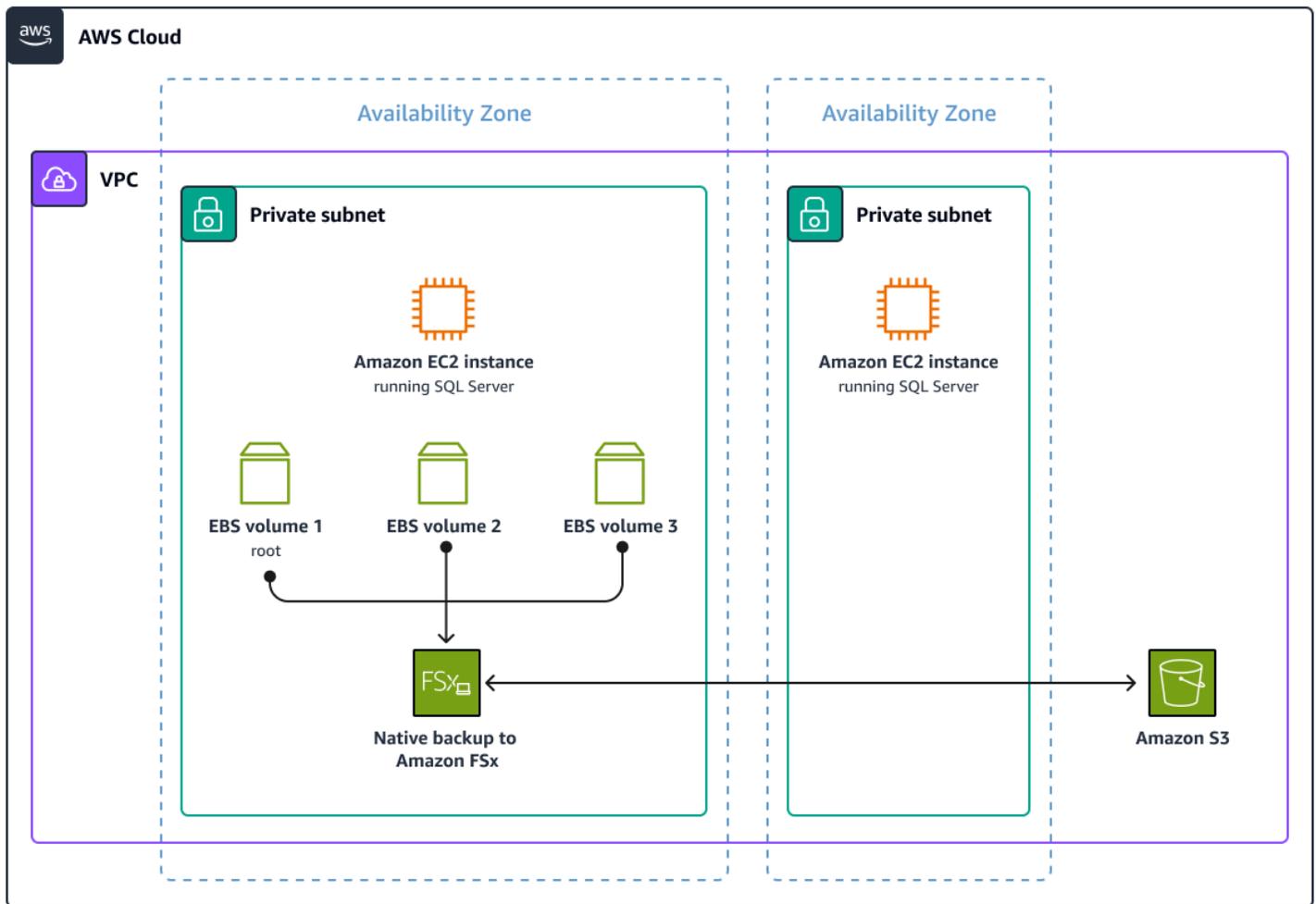
考慮使用原生備份至 EBS 磁碟區的下列限制：

- 必須從 EBS 磁碟區手動將備份移至 Amazon S3。
- 對於大型備份，您必須管理 Amazon EC2 上的磁碟空間。
- 在 EC2 執行個體上，Amazon EBS 輸送量可能成為瓶頸。
- 在 Amazon EBS 上儲存備份需要額外的儲存空間。

## SQL Server 原生備份至 Amazon FSx for Windows File Server

[Amazon FSx for Windows File Server](#) 是全受管的原生 Windows 檔案系統，提供高達 64 TB 的儲存空間，旨在提供快速、可預測且一致的效能。AWS 針對 FSx for Windows File Server 上的[異地同步備份檔案系統部署，引進原生支援](#)。原生支援可讓您更輕鬆地在 上部署 Windows 檔案儲存 AWS，並跨多個可用區域提供高可用性和備援。AWS 也推出對 [SMB 持續可用 \(CA\) 檔案共用的支援](#)。您可以使用 FSx for Windows File Server 做為 SQL Server 資料庫的備份儲存。

下圖顯示 FSx for Windows File Server 的原生 SQL Server 備份架構。



考慮使用原生 SQL Server 備份至 FSx for Windows File Server 的下列優點：

- 您可以將 SQL Server 資料庫備份到 Amazon FSx 檔案共用。
- 您可以備份 SQL Server 執行個體上的單獨資料庫，並還原單獨資料庫，而不必還原整個執行個體。
- 支援分段備份。
- 可以使用 SQL Server Agent 和作業引擎來排程備份作業。
- 相較於 Amazon EBS，執行個體的網路頻寬較高。

考慮使用原生 SQL Server 備份到 FSx for Windows File Server 的下列限制：

- 您必須使用 AWS Backup 或 ，手動將備份從 Amazon FSx 移至 Amazon S3 AWS DataSync。
- 在 Amazon FSx 上，大型備份可能需要額外的磁碟空間管理開銷。
- EC2 執行個體網路輸送量可能是瓶頸。
- 需要額外的儲存空間才能將備份儲存在 FSx for Windows File Server 上。

## SQL Server 備份至 Amazon FSx for NetApp ONTAP

FSx for ONTAP 的快照一律會一致當機，但需要您查詢（或暫停資料庫的輸入/輸出），才能建立應用程式一致快照。您可以使用 NetApp SnapCenter（一種具有特定應用程式外掛程式的協同運作工具，包括 SQL Server）搭配 FSx for ONTAP 來建立應用程式一致性快照，並保護、複寫和複製資料庫，無需額外費用。

### NetApp SnapCenter

NetApp SnapCenter 是應用程式一致資料保護的統一平台。SnapCenter 將快照稱為備份。本指南採用相同的命名慣例。SnapCenter 提供單一窗格，用於管理應用程式一致的備份、還原和複製。您可以為特定資料庫應用程式新增 SnapCenter 外掛程式，以建立應用程式一致的備份。SQL Server 的 SnapCenter 外掛程式提供下列功能，可簡化資料保護工作流程。

- 完整備份和日誌備份的精細程度備份和還原選項
- 就地還原和還原至替代位置

如需 SnapCenter 的詳細資訊，請參閱 AWS 儲存部落格上的[使用 NetApp SnapCenter 搭配 Amazon FSx for NetApp ONTAP 保護 SQL Server 工作負載](#)文章。

### 備份的成本最佳化

下列選項可協助您降低儲存 SQL Server 備份的成本 AWS。

- 在建立備份檔案期間啟用 [SQL Server 壓縮](#)，並將最小的檔案傳送到儲存體。例如，3 : 1 壓縮率表示您要在磁碟空間上節省約 66%。若要查詢這些資料欄，您可以使用下列 Transact-SQL 陳述式：`SELECT backup_size/compressed_backup_size FROM msdb..backupset;`
- 對於前往 S3 儲存貯體的備份，啟用 [Amazon S3 Intelligent-Tiering](#) 儲存類別，將儲存成本降低 30%。
- 對於前往 FSx for Windows File Server 或 FSx for ONTAP 的備份，請使用單一可用區域節省 50% 的成本（相較於使用多個可用區域）。如需定價資訊，請參閱 [Amazon FSx for Windows File Server 定價](#)和 [Amazon FSx for NetApp ONTAP 定價](#)。
- SQL Server 2022 最有效率的選項是直接備份到 Amazon S3。您可以避免 Storage Gateway 來節省額外的成本。

## 備份的基準測試結果

本節根據本指南涵蓋的備份解決方案的效能基準測試結果，比較範例 1 TB 資料庫的成本和效能觀點的下列選項。

- EC2 執行個體規格 – r5d.8xlarge 搭配 Windows Server 2019 和 SQL Server 2019 開發人員版本
- 資料庫規格 – 1 TB 大小，停用 TDE

測試是以 r5d.8xlarge 執行個體和 1 TB SQL Server 資料庫做為來源來執行。來源系統是根據最佳實務設定，而來源資料庫包含四個資料檔案（每個 250 GB）和一個分散在個別 gp3 磁碟區的日誌檔案 (50 GB)。SQL Server 原生BACKUP命令包括寫入 10 個備份檔案、使用壓縮來最佳化備份效能，並減少透過網路傳送並寫入目標的資料量。在所有測試案例中，儲存效能是瓶頸。

對於這些類型的測試，有各種各樣的可能組態。此測試著重於最佳化效能、成本、可擴展性和實際使用案例。下表顯示針對備份目標選項擷取的效能指標。

| 備份選項   | Level | 執行持續時間 (Appx)   | 備份速率       | 每月成本 USD*                |
|--|-------|-----------------|------------|--------------------------|
| 本機 EBS st1 HDD 的原生備份，2 TB                                | 資料庫   | 最低 00 : 30 : 46 | 554.7 Mbps | 92.16 美元                 |
| 本機 EBS SSD gp3 的原生備份，2 TB                                | 資料庫   | 最低 00 : 22 : 00 | 512 Mbps   | 193.84 美元                |
| 原生備份至 FSx for Windows File Server HDD，2 TB @512 Mbps 輸送量 | 資料庫   | 最低 00 : 20 : 58 | 814.0 Mbps | <a href="#">1,146 美元</a> |
| 原生備份至 FSx for Windows File Server SSD，2 TB @512 Mbps 輸送量 | 資料庫   | 最低 00 : 20 : 00 | 814.0 Mbps | <a href="#">1,326 美元</a> |

| 備份選項  | Level   | 執行持續時間 (Appx)                     | 備份速率       | 每月成本 USD*   |
|---|---------|-----------------------------------|------------|---|
| 原生備份至具有 2 TB gp3 的 S3 檔案閘道 m6i.4xlarge (16 vCPU, 64 GB) | 資料庫     | 00 : 23 : 20 分鐘                   | 731.5 Mbps | 470.42 美元   |
| EBS VSS 快照  | EBS 磁碟區 | 00 : 00 : 02 秒<br>00 : 00 : 53 秒  | 不適用快照      | <a href="#">51 美元</a>                             |
| AWS Backup (AMI 備份)                                     | AMI     | 00 : 00 : 04 秒<br>最低 00 : 08 : 00 | 不適用快照      | <a href="#">75 美元</a>                             |
| 直接原生 SQL Server 備份至 Amazon S3 (SQL Server 2022)         | 資料庫     | 最低 00 : 12 : 00                   | 731.5 Mbps | <a href="#">前 50 TB/月, 每 GB \$0.023/月 \$23.55</a> |
| FSx for ONTAP 的原生備份 (使用 SnapCenter)                     | 資料庫     | –                                 | –          | <a href="#">440.20 美元</a>                         |

上表假設下列事項：

- 不包括資料傳輸和 Amazon S3 成本。
- 儲存體價格包含在執行個體定價中。
- 成本以 us-east-1 區域為基礎。
- 輸送量和 IOPS 成長 10%，而多個備份的整體變化率為當月的 10%。

測試結果顯示最快的選項是原生 SQL Server 資料庫備份至 FSx for Windows File Server。對 Storage Gateway 和本機連接的 EBS 磁碟區的備份是更具成本效益的選項，但效能較慢。對於伺服器層級備份 (AMI)，建議使用 AWS Backup 來獲得最佳效能、成本和管理性。

## 成本最佳化建議

了解在 Amazon EC2 上備份 SQL Server 的可能解決方案，是保護資料、確保您符合備份需求，以及制定計畫以從關鍵事件復原的關鍵。本節中探索的備份和還原 SQL Server 執行個體和資料庫的不同方式，可協助您設計備份和還原策略，以保護資料並滿足組織的需求。

本節涵蓋下列備份選項：

- 壓縮
- Amazon S3 Intelligent-Tiering
- 單一可用區域
- 備份至 URL

每個選項所提供的指引都是高階。如果您想要在您的組織中實作任何這些建議，我們建議您聯絡您的客戶團隊。然後，團隊可以與 Microsoft 專家 SA 互動以領導對話。您也可以傳送電子郵件至 [optimize-microsoft@amazon.com](mailto:optimize-microsoft@amazon.com) 來聯絡。

總而言之，我們建議下列事項：

- 如果您使用的是 SQL Server 2022，則備份到 Amazon S3 是最符合成本效益的選項。
- 如果您使用的是 SQL Server 2019 和舊版 SQL Server，請考慮將 Amazon S3 支援的 Storage Gateway 備份為最具成本效益的選項。

### 壓縮

壓縮的目標是減少每個備份耗用的儲存空間，這有利於各種儲存選項。您必須在 SQL Server [執行個體層級啟用 SQL Server](#) 備份的壓縮。下列範例示範如何使用備份資料庫新增壓縮關鍵字：

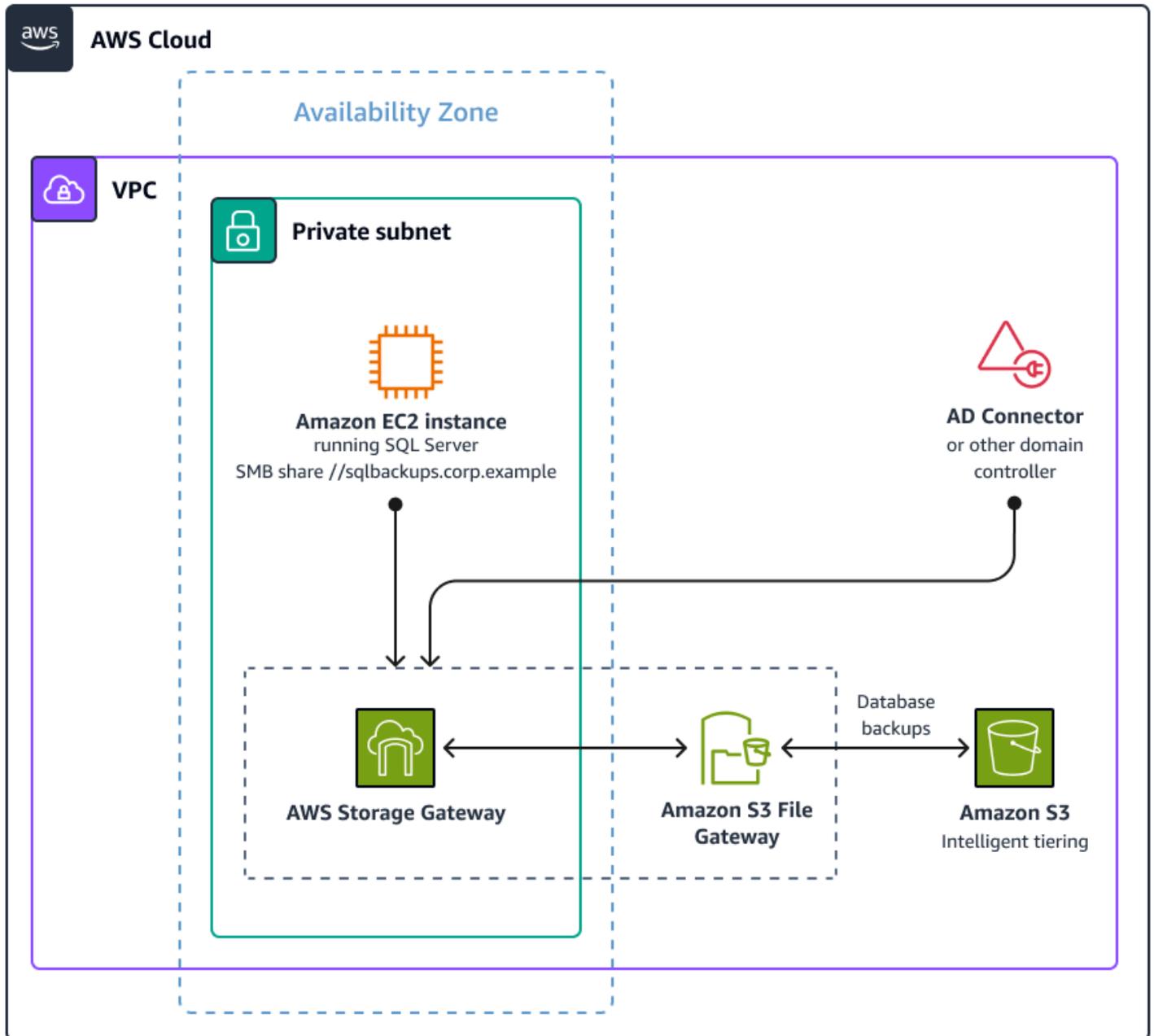
```
BACKUP DATABASE <database_name> TO DISK WITH COMPRESSION (ALGORITHM =
QAT_DEFLATE)
```

### Amazon S3 Intelligent-Tiering

對於前往 Amazon S3 儲存貯體的備份，您可以啟用 [Amazon S3 Intelligent-Tiering](#) 做為 Amazon S3 File Gateway [儲存類別](#)。這可降低高達 30% 的儲存成本。然後，您可以使用可與 [Active Directory 網域](#) 整合的 SMB 檔案共用，將 S3 檔案閘道掛載到您的 SQL 伺服器。這為您提供共享的存取控制、利用現有服務帳戶的能力，以及使用常見 Microsoft 聚焦檔案通訊協定存取 Amazon S3。對於可能沒有

直接連線至網域控制站的帳戶，您可以使用 [Active Directory Connector](#) 來促進與內部部署或雲端中 Active Directory 的通訊。若要在閘道上設定 Active Directory 設定，您必須指定網域控制器的 Active Directory Connector IPs，以將請求代理至 Active Directory。

下圖顯示以 S3 Intelligent-Tiering 為基礎的解決方案架構。



根據預設，寫入 S3 儲存貯體的備份檔案會使用 Standard 層。若要將備份檔案從標準層轉換為 S3 Intelligent-Tiering，您必須 [建立生命週期規則](#)。您也可以使用 [AWS Management Console](#) 來啟用 S3 Intelligent-Tiering。如需詳細資訊，請參閱 文件中的 AWS [開始使用 Amazon S3 Intelligent-Tiering](#)。

## 單一可用區域

若要建立單一可用區域檔案系統，請在[建立 FSx for Windows File Server 檔案系統](#)時選擇單一可用區域選項。Amazon FSx 也會每天使用 Windows Volume Shadow Copy Service 對您的檔案系統進行高耐用性備份（存放在 Amazon S3 中），並可讓您隨時進行額外的備份。請記住使用單一可用區域的一些問題。例如，如果佈建檔案系統的受影響可用區域一次中斷數小時，則無法存取 SMB 檔案共用。如果您需要存取資料，則必須從來源區域內可用可用區域中的備份還原資料。如需詳細資訊，請參閱本指南的[使用單一可用區域](#)一節。

## 備份至 URL

針對 SQL Server 2022，[備份至 URL](#) 功能允許直接備份至 Amazon S3。這是在中執行 SQL Server 2022 的理想備份方法，AWS 因為您在儲存層取得完整的功能集 Amazon S3，並移除先前版本所需的設備成本 AWS Storage Gateway，以促進此功能。實作此功能時，需要考慮兩個主要成本：資料傳輸成本和選擇的 S3 儲存體方案。如果您想要 Amazon S3 的原生災難復原功能，則必須考量[該跨區域複寫](#)會產生跨區域[資料輸出成本](#)。若要進一步了解如何設定此選項，請參閱部落格 Microsoft 工作負載 AWS 上的將 [SQL Server 資料庫備份至 Amazon S3](#) 文章。

## 其他資源

- [Amazon EC2 上 SQL Server 的備份和還原選項](#) (AWS 方案指引)
- [使用的 Amazon RDS Point-in-time復原和持續備份 AWS Backup](#) (AWS 儲存部落格)
- [使用 NetApp SnapCenter 搭配 Amazon FSx for NetApp ONTAP 保護您的 SQL Server 工作負載](#) (AWS 儲存部落格)
- [開始使用 Amazon S3 Intelligent-Tiering](#) (AWS 入門資源中心)
- [Amazon RDS for SQL Server 的備份和還原策略](#) (AWS 資料庫部落格)
- [將內部部署 Microsoft SQL Server 資料庫遷移至 Amazon EC2](#) (AWS 方案指引)
- [在 Amazon EC2 上部署 Microsoft SQL Server 的最佳實務](#) (AWS 白皮書)

## 現代化 SQL Server 資料庫

### 概觀

如果您開始將舊版資料庫現代化以實現可擴展性、效能和成本最佳化，您可能會面臨 SQL Server 等商業資料庫的挑戰。商業資料庫非常昂貴、鎖定客戶，並提供懲罰性授權條款。本節提供從 SQL Server 遷移和現代化至開放原始碼資料庫的選項的高階概觀，以及為您的工作負載選擇最佳選項的相關資訊。

您可以將 SQL Server 資料庫重構為開放原始碼資料庫，例如 Amazon Aurora PostgreSQL，以節省 Windows 和 SQL Server 授權成本。Aurora 等雲端原生現代資料庫結合了開放原始碼資料庫的彈性和低成本，以及商業資料庫的強大企業級功能。如果您有可變工作負載或多租戶工作負載，您也可以遷移至 [Aurora 無伺服器 V2](#)。視工作負載特性而定，這可將成本降低 90%。此外，AWS 提供 [Babelfish for Aurora PostgreSQL](#) 等功能、[AWS Schema Conversion Tool \(AWS SCT\)](#) 等工具，以及 [AWS Database Migration Service \(AWS DMS\)](#) 等服務，以簡化 SQL Server 資料庫的遷移和現代化 AWS。

## 資料庫產品

從 Windows 上的 SQL Server 遷移到開放原始碼資料庫，例如 Amazon Aurora、Amazon RDS for MySQL 或 Amazon RDS for PostgreSQL，可以大幅節省成本，而不會犧牲效能或功能。考慮下列各項：

- 從 Amazon EC2 上的 SQL Server Enterprise Edition 切換到 Amazon RDS for PostgreSQL 或 Amazon RDS for MySQL 可以節省高達 80% 的成本。
- 從 Amazon EC2 上的 SQL Server Enterprise Edition 切換至 Amazon Aurora PostgreSQL 相容版本或 Amazon Aurora MySQL 相容版本，可節省高達 70% 的成本。

對於傳統資料庫工作負載，Amazon RDS for PostgreSQL 和 Amazon RDS for MySQL 地址要求，並為關聯式資料庫提供經濟實惠的解決方案。Aurora 新增了許多先前僅限於昂貴商業廠商的可用性和效能功能。Aurora 中的彈性功能是增加的成本。不過，相較於其他商業廠商的類似功能，Aurora 的彈性成本仍然比相同類型功能的商業軟體費用便宜。Aurora 架構經過最佳化，相較於標準 MySQL 和 PostgreSQL 部署，可大幅改善效能。

由於 Aurora 與開放原始碼 PostgreSQL 和 MySQL 資料庫相容，因此具有額外的可攜性優勢。無論最佳選項是 Amazon RDS for PostgreSQL、Amazon RDS for MySQL 或 Aurora，都必須了解業務需求並將必要的功能映射至最佳選項。

## Amazon RDS 和 Aurora 比較

下表摘要說明 Amazon RDS 和 Amazon Aurora 之間的主要差異。

| 類別   | Amazon RDS for PostgreSQL 或 Amazon RDS for MySQL | Aurora PostgreSQL 或 Aurora MySQL |
|------|--|----------------------------------|
| 效能   | 良好的效能  | 效能提升 3 倍以上                       |
| 容錯移轉 | 通常為 60–120 秒*                                    | 通常為 30 秒                         |

| 類別         | Amazon RDS for PostgreSQL 或 Amazon RDS for MySQL | Aurora PostgreSQL 或 Aurora MySQL |
|------------|--|----------------------------------|
| 可擴展性       | 最多 5 個僅供讀取複本<br>以秒為單位的延遲                         | 最多 15 個僅供讀取複本<br>以毫秒為單位的延遲       |
| 儲存         | 高達 64 TB   | 高達 128 TB                        |
| 儲存 HA      | 具有一或兩個待命的異地同步備份，每個都具有資料庫副本                       | 根據預設，跨 3 個可用區域 6 個資料複本           |
| 備份         | 每日快照和日誌備份  | 連續、非同步備份至 Amazon S3              |
| Aurora 的創新 | NA   | 100 GB<br>快速資料庫複製                |
|            | 自動擴展僅供讀取複本                                       |                                  |
|            | 查詢計劃管理   |                                  |
|            | Aurora Serverless                                |                                  |
|            | 具有全域資料庫的跨區域複本                                    |                                  |
|            | 叢集快取管理**   |                                  |
|            | 平行查詢   |                                  |
|            | 資料庫活動串流  |                                  |

\*大型交易可能會增加容錯移轉時間

\*\*可在 Aurora PostgreSQL 中使用

下表顯示本節涵蓋的不同資料庫服務的預估每月成本。

| 資料庫服務  | 每月成本 USD* | AWS 定價計算工具 (需要 AWS 帳戶) |
|--|-----------|------------------------|
| Amazon RDS for SQL Server Enterprise Edition | 3,750 美元  | <a href="#">估算</a>     |
| Amazon RDS for SQL Server Standard 版         | 2,318 美元  | <a href="#">估算</a>     |
| Amazon EC2 上的 SQL Server Enterprise Edition  | 2,835 美元  | <a href="#">估算</a>     |
| Amazon EC2 上的 SQL Server Standard 版          | 1,345 美元  | <a href="#">估算</a>     |
| Amazon RDS for PostgreSQL                    | 742 美元    | <a href="#">估算</a>     |
| Amazon RDS for MySQL                         | 712 美元    | <a href="#">估算</a>     |
| Aurora PostgreSQL                            | 1,032 美元  | <a href="#">估算</a>     |
| Aurora MySQL                                 | 1,031 美元  | <a href="#">估算</a>     |

\* 儲存體價格包含在執行個體定價中。成本以 us-east-1 區域為基礎。輸送量和 IOPS 是假設。計算適用於 r6i.2xlarge 和 r6g.2xlarge 執行個體。

## 成本最佳化建議

異質資料庫遷移通常需要將資料庫結構描述從來源轉換為目標資料庫引擎，並將資料從來源遷移至目標資料庫。遷移的第一步是評估 SQL Server 結構描述和程式碼物件並將其轉換為目標資料庫引擎。

您可以使用 [AWS Schema Conversion Tool \(AWS SCT\)](#) 來評估資料庫與各種目標開放原始碼資料庫選項的相容性，例如 Amazon RDS for MySQL 或 Amazon RDS for PostgreSQL、Aurora MySQL 和 PostgreSQL。您也可以使用 Babelfish Compass 工具來評估與 Babelfish for Aurora PostgreSQL 的相容性。這可讓 AWS SCT 和 Compass 強大的工具了解在決定遷移策略之前涉及的前期工作。如果您決定繼續，會自動 AWS SCT 執行結構描述所需的變更。Babelfish Compass 背後的核心理念是允許 SQL 資料庫移至 Aurora，無需或很少修改。Compass 將評估現有的 SQL 資料庫，以確定是否可以完成此操作。如此一來，在將資料從 SQL Server 遷移到 Aurora 之前，就會知道結果。

AWS SCT 會將資料庫結構描述和程式碼的轉換和遷移自動化到目標資料庫引擎。您可以使用 Babelfish for Aurora PostgreSQL 將資料庫和應用程式從 SQL Server 遷移至 Aurora PostgreSQL，而不需要變更結構描述或變更最少。這可以加速遷移。

遷移結構描述之後，您可以使用 AWS DMS 遷移資料。AWS DMS 可以執行完整資料載入並複寫變更，以將停機時間降至最低來執行遷移。

本節會更詳細地探索下列工具：

- AWS Schema Conversion Tool
- Babelfish for Aurora PostgreSQL
- Babelfish Compass
- AWS Database Migration Service

## AWS Schema Conversion Tool

您可以使用 AWS SCT 來評估現有的 SQL Server 資料庫，並評估與 Amazon RDS 或 Aurora 的相容性。若要簡化遷移程序，您也可以使用在異質資料庫遷移中，將結構描述從一個資料庫引擎 AWS SCT 轉換為另一個資料庫引擎。您可以使用 AWS SCT 來評估您的應用程式，並為撰寫 C#、C++、Java 和其他語言的應用程式轉換內嵌應用程式程式碼。如需詳細資訊，請參閱文件中的 [AWS SCT 使用 轉換應用程式 SQL AWS SCT](#)。

AWS SCT 是支援許多資料庫來源的免費 AWS 工具。若要使用 AWS SCT，請將其指向來源資料庫，然後執行評估。然後，會 [AWS SCT](#) 評估結構描述並產生評估報告。評估報告包括執行摘要、複雜性和遷移工作、適合的目標資料庫引擎，以及轉換的建議。若要下載 AWS SCT，請參閱文件中的 [AWS SCT 安裝、驗證和更新 AWS SCT](#)。

下表顯示 AWS SCT 產生的範例執行摘要，以顯示將資料庫變更為不同目標平台所涉及的複雜性。

| 目標平台                 | 自動或最小變更  |         | 複雜動作 |        |       |          |    |
|----------------------|----------|---------|------|--------|-------|----------|----|
|                      | 儲存物件     | 程式碼物件   | 轉換動作 | 儲存物件   | 程式碼物件 |          |    |
| Amazon RDS for MySQL | 60 (98%) | 8 (35%) | 42   | 1 (2%) | 1     | 15 (65%) | 56 |

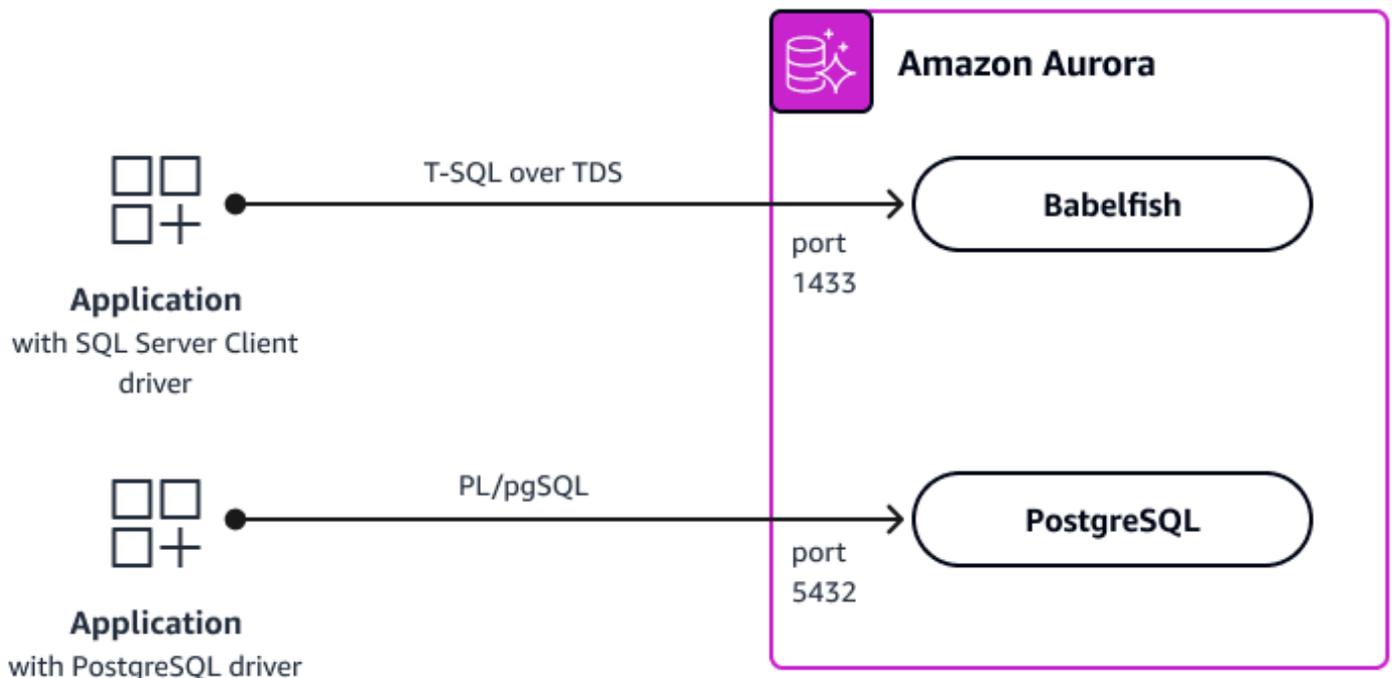
|   |           |           |     |        |   |          |    |
|---|-----------|-----------|-----|--------|---|----------|----|
| Amazon Aurora MySQL-Compatible Edition      | 60 (98%)  | 8 (35%)   | 42  | 1 (2%) | 1 | 15 (65%) | 56 |
| Amazon RDS for PostgreSQL                   | 60 (98%)  | 12 (52%)  | 54  | 1 (2%) | 1 | 11 (48%) | 26 |
| Amazon Aurora PostgreSQL-Compatible Edition | 60 (98%)  | 12 (52%)  | 54  | 1 (2%) | 1 | 11 (48%) | 26 |
| Amazon RDS for MariaDB                      | 60 (98%)  | 7 (30%)   | 42  | 1 (2%) | 1 | 16 (70%) | 58 |
| Amazon Redshift                             | 61 (100%) | 9 (39%)   | 124 | 0 (0%) | 0 | 14 (61%) | 25 |
| AWS Glue                                    | 0 (0%)    | 17 (100%) | 0   | 0 (0%) | 0 | 0 (0%)   | 0  |
| Babelfish                                   | 59 (97%)  | 10 (45%)  | 20  | 2 (3%) | 2 | 12 (55%) | 30 |

AWS SCT 報告也提供無法自動轉換之結構描述元素的詳細資訊。您可以參考[AWS 遷移手冊](#)來關閉 AWS SCT 轉換差距並最佳化目標結構描述。有許多資料庫遷移手冊可協助異質遷移。

## Babelfish for Aurora PostgreSQL

Babelfish for Aurora PostgreSQL 擴展了 Aurora PostgreSQL，能夠接受來自 SQL Server 用戶端的資料庫連線。Babelfish 可讓最初為 SQL Server 建置的應用程式直接與 Aurora PostgreSQL 搭配使用，幾乎不需要變更程式碼，也不會變更資料庫驅動程式。Babelfish 將 Aurora PostgreSQL 轉換為雙語，以便 Aurora PostgreSQL 可以使用 T-SQL 和 PL/pgSQL 語言。Babelfish 會將從 SQL Server 遷移到 Aurora PostgreSQL 的工作量降至最低。這可加速遷移、將風險降至最低，並大幅降低遷移成本。您可以繼續使用 T-SQL 後遷移，但也可以[選擇使用 PostgreSQL 原生工具](#)進行開發。

下圖說明使用 T-SQL 的應用程式如何連接到 SQL Server 中的預設連接埠 1433，並使用 Babelfish 翻譯器與 Aurora PostgreSQL 資料庫通訊，而使用 PL/pgSQL 的應用程式可以使用 Aurora PostgreSQL 中的預設連接埠 5432 直接並同時連接到 Aurora PostgreSQL 資料庫。



Babelfish 不支援特定 SQL Server T-SQL 功能。因此，Amazon 提供評估工具，對您的 SQL 陳述式進行 line-by-line 分析，並判斷 Babelfish 是否不支援其中任何一項。

Babelfish 評估有兩個選項。AWS SCT 可以評估 SQL Server 資料庫與 Babelfish 的相容性。另一個選項是 Babelfish Compass 工具，這是建議的解決方案，因為 Compass 工具會根據 Babelfish for Aurora PostgreSQL 的新版本進行更新。

### Babelfish Compass

[Babelfish Compass](#) 是免費下載的工具，符合 Babelfish for Aurora PostgreSQL 的最新版本。相反地，AWS SCT 會在一段時間後支援較新的 Babelfish 版本。[Babelfish Compass](#) 會針對 SQL Server 資

料庫結構描述執行。您也可以使用 SQL Server Management Studio (SSMS) 等工具來擷取來源 SQL Server 資料庫結構描述。然後，您可以透過 Babelfish Compass 執行結構描述。這會產生報告，詳細說明 SQL Server 結構描述與 Babelfish 的相容性，以及在遷移之前是否需要任何變更。Babelfish Compass 工具也可以自動化許多這些變更，最終加速遷移。

評估和變更完成後，您可以使用 SSMS 或 sqlcmd 等 SQL Server 原生工具，將結構描述遷移至 Aurora PostgreSQL。如需說明，請參閱 AWS 資料庫部落格上的[使用 Babelfish 文章從 SQL Server 遷移至 Amazon Aurora](#)。

## AWS Database Migration Service

遷移結構描述之後，您可以使用 AWS Database Migration Service (AWS DMS) 以最短 AWS 的停機時間將資料遷移至。AWS DMS 不僅會執行完整的資料載入，還會在來源系統啟動並執行時，將變更從來源複寫至目的地。在來源和目標資料庫都同步之後，切換活動可能會發生，其中應用程式指向完成遷移的目標資料庫。AWS DMS 目前，只會針對 Aurora PostgreSQL 目標使用 Babelfish 執行完整資料載入，而不會複寫變更。如需詳細資訊，請參閱 AWS DMS 文件中的[使用 Babelfish 做為的目標 AWS Database Migration Service](#)。

AWS DMS 可以同時執行同質（跨同一資料庫引擎）和異質（跨不同資料庫引擎）遷移。AWS DMS 支援許多來源和目的地資料庫引擎。如需詳細資訊，請參閱 資料庫部落格中的[使用文章將 SQL Server 資料庫遷移至 Amazon RDS for SQL Server AWS DMS](#)。AWS

## 其他資源

- [再見 Microsoft SQL Server、Hello Babelfish](#) (AWS 新聞部落格)
- [使用 AWS Schema Conversion Tool CLI 轉換資料庫結構描述和應用程式 SQL](#) (AWS 資料庫部落格)
- [使用從 欄位學到的最佳實務和經驗，將 SQL Server 遷移至 Amazon Aurora PostgreSQL](#) (AWS 資料庫部落格)
- [驗證從 Microsoft SQL Server 遷移到 Amazon RDS for PostgreSQL 和 Amazon Aurora PostgreSQL 的資料庫物件](#) (AWS 資料庫部落格)

## 最佳化 SQL Server 的儲存

### 概觀

本節著重於 EC2 工作負載上 SQL Server 的 Amazon Elastic Block Store (Amazon EBS) SSD 儲存體的成本最佳化。

您有多種用於部署和執行 SQL Server 工作負載的儲存選項 AWS。選取正確的儲存體應以用途、架構、耐用性、效能、容量和成本為基礎。執行 SQL Server 工作負載 AWS 的客戶通常會使用 Amazon EBS、NVMe、Amazon FSx 和 Amazon Simple Storage Service (Amazon S3) 儲存體的組合。

Amazon EBS 是連接至 EC2 運算執行個體的網路連接儲存體，用於存放和處理一般作業系統、應用程式、資料庫和備份檔案。Amazon EBS 固態硬碟 (SSD) 儲存包含一般用途 SSD (gp2 和 gp3) 和佈建 IOPS SSD (io1、io2 和 io2BX)。考慮下列各項：

- 有些 EC2 執行個體，例如 r5d，具有實際連接到主機執行個體的本機 NVMe SSDs。這些磁碟區提供區塊層級儲存，常用於 SQL Server tempdb 或緩衝集區延伸。
- Amazon FSx for Windows File Server 是全受管檔案儲存服務，而 Amazon FSx for NetApp ONTAP 是建置在 NetApp 熱門 ONTAP 檔案系統上的全受管共用儲存。Amazon FSx 經常用於在高可用性、SQL Server 容錯移轉叢集執行個體 (FCI) 組態中執行 SQL Server 工作負載。此解決方案託管 SQL Server 資料和日誌檔案，可降低 EC2 執行個體的 EBS 效能需求。
- Amazon S3 是一種物件儲存服務，提供業界領先的可擴展性、資料可用性、安全性和效能。您可以在 Amazon S3 上存放 SQL Server 原生備份檔案、AMIs、EBS 快照、應用程式日誌等。

## Amazon EBS 的 SSD 儲存類型、效能和成本

Amazon EBS 的 SSD 儲存成本通常會隨著耐用性和效能的增加而增加。儲存目前有五種磁碟區類型，每個類型都有[自己的唯一效能指標](#)。如需 SSD 後端磁碟區的使用案例和特性摘要，請參閱 Amazon EBS 文件的[固態硬碟 \(SSD\) 磁碟區](#)一節中的表格。

您可以使用 Amazon CloudWatch 來監控 SSD 效能、擷取趨勢資料，以及在達到特定閾值時設定警示。如果您在上執行 SQL Server 工作負載 AWS，請考慮啟用[詳細的監控和部署 CloudWatch 自訂指標](#)，以擷取詳細的磁碟效能指標，例如磁碟延遲、IOPS、輸送量、磁碟佇列長度、已使用容量與可用容量等。您可以使用這些 CloudWatch 效能指標來識別佈建不足和過度佈建的儲存，並提供歷史資料點來準確定義儲存需求。

Amazon EBS 的 SSD 儲存成本也會因配置的容量而有所不同。下表顯示不同磁碟區的比較。所有磁碟區類型都有 1 TB 的容量和類似的效能組態。

| 磁碟區類型 | 最大 IOPS (16 KiB I/O) | 最大輸送量 (128 KiB I/O) | 每 1TB 的價格 | 節省成本百分比 |
|-------|----------------------|---------------------|-----------|---------|
| gp2   | 3,000                | 250                 | 102.40 美元 |         |
| gp3   | 3,000                | 250                 | 86.92 美元  | 15%     |

| 磁碟區類型 | 最大 IOPS (16 KiB I/O) | 最大輸送量 (128 KiB I/O) | 每 1TB 的價格 | 節省成本百分比 |
|-------|----------------------|---------------------|-----------|---------|
| io1   | 16,000               | 500                 | 1,168 美元  |         |
| io2   | 16,000               | 500                 | 1,168 美元  |         |
| gp3   | 16,000               | 500                 | 146.92 美元 | 87%     |
| io2bx | 16,000               | 4,000               | 1,168 美元  |         |
| gp3   | 16,000               | 1,000               | 181.92 美元 | 84%     |

### Note

上表中的效能和成本指標是根據來自的[預估](#)，以每個磁碟區為單位 AWS 定價計算工具。AWS 帳戶需要才能存取中的預估值 AWS 定價計算工具。

Amazon EBS SSD gp3 磁碟區以低成本提供絕佳的效能。對於需要低於 16,000 IOPS 和 500 MiBps 輸送量的工作負載，如果您透過 io1 或 io2 磁碟區選擇 gp3 磁碟區，則最多可節省 87%。

io2 Block Express (io2BX) 磁碟區比一般 io2 磁碟區提供更高的效能。在 16,000 IOPS 時，io1 或 io2 磁碟區只能傳輸 500 MiBps，而 io2BX 磁碟區最多可設定 4,000 MiBps 傳輸量。相較於 io1 和 io2 磁碟區，io2BX 磁碟區以完全相同的價格提供介於 16,000 到 64,000 IOPS 之間的輸送量四倍以上。將一般 io2 磁碟區連接到 io2BX io2BX-supported 磁碟區。EC2 如需 io2BX-supported EC2 執行個體清單，請參閱 Amazon EBS 文件中的[佈建 IOPS SSD 磁碟區](#)。部署新儲存之前，您可以使用 [AWS 定價計算工具](#) 來估算每月成本，並根據持久性、效能和容量之間的權衡，了解對成本的影響。

## Amazon EBS 的一般 SSD 成本最佳化

我們建議您評估要儲存的內容，並確保您使用的是正確的儲存類型和類別。例如，Amazon S3 提供絕佳的價位、內建生命週期政策和複寫選項，非常適合 SQL Server 備份。SQL Server 2022 能夠直接備份到 Amazon S3，而舊版的 SQL Server 依賴原生本機備份。如果您正在執行舊版 SQL Server，請考慮備份至 Amazon EBS HDD 磁碟區，然後將備份複製到 Amazon S3。此解決方案可以節省 53%，而不是使用 gp3 磁碟區進行備份。

下表顯示 Amazon EBS gp3、Amazon EBS HDD st1 和 Amazon S3 上 1 TB 儲存體的價格差異。

| 儲存體類型                   | 容量   | 價格 pm    |
|-------------------------|------|----------|
| EBS gp3 500 MiBps       | 1 TB | 96.92 美元 |
| EBS st1 爆量 500 MiBps    |      | 46.08 美元 |
| S3 Standard             |      | 23.55 美元 |
| S3 Standard (不常存取)      |      | 12.80 美元 |
| S3 Glacier Deep Archive |      | 1.03 美元  |

### Note

上表中的成本指標是根據中的 [預估](#) AWS 定價計算工具值。AWS 帳戶需要才能存取中的預估 AWS 定價計算工具。

我們建議您考慮下列事項：

- 啟用詳細的監控和部署 CloudWatch 自訂指標，以準確擷取其儲存效能需求。
- 將 Amazon EBS 儲存體從 gp2 升級到 gp3，以降低成本、提高靈活性並改善效能。
- 將 Amazon EBS 儲存體從 io1 升級到 io2，以提高耐用性和效能彈性。
- 盡可能使用 io2BX 而非 io1 或 io2，以提高耐用性和效能。
- 選擇儲存體時，請考慮 mix-and-match 方法，以協助減少容量需求和高效能磁碟區的成本。例如，您可以將低成本 gp3 磁碟區用於根磁碟區（作業系統）、SQL Server 安裝、系統資料庫（不包括 tempdb），以及效能較低的使用者資料庫。這有助於降低 io2 磁碟區的容量和成本，這可以專用於高效能使用者資料庫。
- 如果您要在上託管 SQL Server 資料庫 AWS，建議您在每個資料庫使用多個 SQL Server 資料檔案。這可讓有機會將讀取/寫入工作負載分散到多個磁碟區，進而降低每個磁碟區的效能和容量需求，進而降低成本。
- 即使生產工作負載需要更高效能的儲存，例如 io1 或 io2/io2BX，請考慮非生產工作負載的 gp3 磁碟區，以協助降低成本。
- 隨著時間的推移追蹤和趨勢儲存使用率，以輕鬆識別使用量峰值和意外成本。
- [AWS Compute Optimizer](#) 用於根據實際使用率向上或向下擴展 EBS 磁碟區的建議。

- 使用的彈性 AWS 來調整 Amazon EBS SSD 磁碟區的效能和容量需求。與內部部署環境不同，您不需要為未來的工作負載過度佈建儲存效能和容量。您可以將現有的 SQL Server 工作負載遷移至 AWS，並視需要調整效能或容量，同時保持資料庫在線上。

## 其他資源

- [Amazon EBS 磁碟區類型](#) (Amazon EBS 文件)
- [Amazon Elastic Block Store \(Amazon EBS\)](#) (Amazon EBS 文件)
- [佈建 IOPS SSD 磁碟區](#) (Amazon EBS 文件)
- [SSD 執行個體存放區磁碟區](#) (Amazon EC2 文件)
- [Amazon EBS 的 Amazon CloudWatch 指標](#) (Amazon EBS 文件)
- [Amazon EC2 儲存最佳化執行個體的規格](#) (Amazon EC2 文件)
- [使用 NetApp SnapCenter 搭配 Amazon FSx for NetApp ONTAP 保護您的 SQL Server 工作負載](#) (AWS 儲存部落格)
- [Amazon EC2 常見問答集](#) (AWS 產品頁面)

## 使用 Compute Optimizer 最佳化 SQL Server 授權

如何使用最佳化 SQL Server 授權的指引 AWS Compute Optimizer。

### 概觀

[AWS Compute Optimizer](#) 可以在 Amazon Elastic Compute Cloud (Amazon EC2) 上建議 Microsoft SQL Server 工作負載的授權最佳化機會。Compute Optimizer 可以提供自動化建議，以降低授權成本。來自 Compute Optimizer 的建議會列在具有 Microsoft SQL Server 授權的每個 EC2 執行個體旁。提供的資訊包括建議的儲存機會、EC2 執行個體隨需價格，以及每小時自帶授權 (BYOL) 價格。此資訊可協助您決定是否應該降級授權版本。

Compute Optimizer 會根據推斷的工作負載類型自動探索 Amazon EC2 上的 SQL Server 執行個體。若要檢視授權建議，您可以在 Compute Optimizer 中選取 SQL Server 執行個體，然後使用唯讀資料庫憑證向 [Amazon CloudWatch Application Insights](#) 進行驗證。Compute Optimizer 會分析您是否使用任何 SQL Server Enterprise Edition 功能。如果未使用 Enterprise Edition 功能，Compute Optimizer 建議您降級至 Standard Edition，以降低授權成本。

您也可以使用 Compute Optimizer 為執行 SQL Server 工作負載的 Amazon EC2 執行個體提出調整大小建議。如需詳細資訊，請參閱本指南中的 [使用 Compute Optimizer 最佳化 SQL Server 大小](#)。

## 成本最佳化建議

Compute Optimizer 中的授權建議可協助您評估在 Microsoft SQL Server 中使用的功能，並為工作負載選擇最具成本效益的版本。SQL Server Enterprise Edition 比 Standard Edition 更昂貴。如需詳細資訊，請參閱本指南中的[比較 SQL Server 版本](#)，並參閱 Microsoft [網站上的 SQL Server 2022 定價](#)。投資時間來設定 Compute Optimizer 以評估 SQL Server 機群並提供建議，可以大幅降低授權成本。

授權詳細資訊頁面提供下列資訊：

- 使用資料表來比較您目前的授權設定（例如版本、模型和執行個體核心數量）與 Compute Optimizer 建議。
- 使用使用率圖表來檢閱在分析期間使用的企業版功能數量。

如需詳細資訊，請參閱 Compute Optimizer 文件中的[檢視商業軟體授權建議的詳細資訊](#)。

## 設定運算最佳化工具

Compute Optimizer 會使用 `mssql_enterprise_features_used` 指標來分析商業軟體授權。如需此指標的詳細資訊，請參閱[商業軟體授權的指標](#)。

1. 請確定您擁有選擇加入 Compute Optimizer 的適當許可。如需詳細資訊，請參閱下列內容：
  - [選擇加入 Compute Optimizer 的政策](#)
  - [授予獨立運算最佳化工具存取權的政策 AWS 帳戶](#)
  - [授予組織管理帳戶 Compute Optimizer 存取權的政策](#)
2. 連接 CloudWatch Application Insights 所需的執行個體角色和政策。如需說明，請參閱[政策以啟用商業軟體授權建議](#)。
3. 使用您的 Microsoft SQL Server 資料庫登入資料啟用 CloudWatch Application Insights。如需說明，請參閱 CloudWatch 文件中的[設定應用程式以進行監控](#)。

### Note

若要產生商業軟體授權的建議，至少需要連續 30 小時的 CloudWatch 指標資料。如需詳細資訊，請參閱 [CloudWatch 指標需求](#)。

4. 使用下列 SQL 查詢來設定 CloudWatch Application Insights 的最低權限存取。

```
GRANT VIEW SERVER STATE TO [LOGIN];
```

```
GRANT VIEW ANY DEFINITION TO [LOGIN];
```

這會啟用新的服務 PrometheusSqlExporterSQL。

5. 從目標 AWS 帳戶 或組織管理帳戶中，選擇加入 Compute Optimizer。如需說明，請參閱[選擇加入您的帳戶](#)。

#### Note

選擇加入後，問題清單和最佳化建議最多可能需要 24 小時才會產生。

6. 在 [Compute Optimizer 主控台](#) 中，選擇導覽窗格中的授權。
7. 在調查結果欄中，搜尋指標調查結果不足的任何執行個體。如果 Compute Optimizer 偵測到 CloudWatch Application Insights 未啟用或許可不足，則會傳回此調查結果。如需詳細資訊，請參閱[尋找原因](#)。執行下列動作來解決這些問題清單：
  - a. 選擇執行個體。
  - b. 新增秘密。
  - c. 確認已連接執行個體角色和政策。
  - d. 選擇啟用授權建議。
8. 在調查結果欄中，搜尋任何具有未最佳化調查結果的執行個體。如果 Compute Optimizer 偵測到您的 Amazon EC2 基礎設施未使用您支付的任何 Microsoft SQL Server 授權功能，則會傳回此調查結果。如需詳細資訊，請參閱[尋找原因](#)。執行下列動作來解決這些問題清單：
  - a. 選擇執行個體。
  - b. 比較目前的授權版本與建議的版本。
  - c. 檢閱目前的授權使用率圖表。
  - d. 如果您想要降級授權，請選擇實作建議。
  - e. 檢閱需求並依照指示降級授權。如果您想要自動化程序，請參閱[使用 AWS Systems Manager 文件降級 SQL Server Enterprise Edition 以降低成本](#) (AWS 部落格)。

## 其他資源

- [使用 降低 Microsoft SQL Server 授權成本 AWS Compute Optimizer](#) (AWS 部落格)
- [什麼是 AWS Compute Optimizer?](#) (AWS 文件)
- [檢視商業軟體授權建議](#) (AWS 文件)
- [降級 Microsoft SQL Server 版本](#) (AWS 文件)

- (AWS) [上的 Microsoft SQL Server AWS](#)
- [上的 Microsoft 授權 AWS](#) (AWS)
- [Microsoft SQL Server 2019 定價](#) (Microsoft)
- [Microsoft SQL Server 2022 定價](#) (Microsoft)

## 使用 Compute Optimizer 最佳化 SQL Server 大小

### 概觀

[AWS Compute Optimizer](#) 協助資料庫管理員 (DBAs) 探索 Amazon Elastic Compute Cloud (Amazon EC2) 上的 Microsoft SQL Server 工作負載，並授權 EC2 執行個體以降低高達 25% 的授權成本。Compute Optimizer [中的推斷工作負載類型](#) 功能使用機器學習 (ML)，並自動偵測可能在 AWS 資源上執行的應用程式。Compute Optimizer 包含 SQL Server 做為推斷工作負載類型的支援。透過使用推斷的工作負載類型功能，您可以根據在 Amazon EC2 執行個體上執行的特定工作負載，精確找出節省成本的機會。

使用此功能，您可以透過 SQL Server 等支援的推斷工作負載類型來分類節省成本的機會。Compute Optimizer 可以自動探索過度佈建的 SQL Server EC2 執行個體。您可以切換到 EC2 主控台來縮減執行個體的大小，這有助於降低授權和基礎設施成本。

您也可以使用 Compute Optimizer 提出 SQL Server 授權建議。如需詳細資訊，請參閱本指南中的 [使用 Compute Optimizer 最佳化 SQL Server 授權](#)。

### 設定運算最佳化工具

如需將 Compute Optimizer 與 SQL Server 推斷工作負載搭配使用的說明，請參閱 [最佳化效能和降低授權成本：利用 AWS Compute Optimizer Amazon EC2 SQL Server 執行個體](#) (AWS 部落格)。您可以選擇加入獨立帳戶、屬於組織成員的帳戶，以及組織的管理帳戶。對於獨立和成員帳戶，選擇加入只會啟用該帳戶的 Compute Optimizer。對於組織管理帳戶，您可以選擇是否僅在該帳戶中啟用 Compute Optimizer，還是為組織的所有成員帳戶啟用。

Compute Optimizer 選擇加入程序會自動建立 AWS Identity and Access Management (IAM) 服務連結角色。如需詳細資訊，請參閱 [使用 AWS Compute Optimizer 的服務連結角色](#)。

Compute Optimizer 會根據 Amazon CloudWatch 指標分析您的資源，例如 CPU、I/O、網路和 Amazon Elastic Block Store (Amazon EBS) 用量。若要產生建議，在過去 14 天內至少需要連續 30 小時的 CloudWatch 指標資料。如果您啟用增強型基礎設施指標功能，則會將使用率指標延長至 93 天。如需詳細資訊，請參閱 Compute Optimizer 文件中的 [CloudWatch 指標需求](#) 和 [增強型基礎設施指標](#)。

Compute Optimizer 會根據 vCPU、記憶體、儲存體、網路、風險和遷移工作，提供與每個選項相關的選項和節省成本。您可以使用 CloudWatch 指標儀表板來分析用於提出建議的資料。使用此資料，您可以授權正在執行 SQL Server 工作負載的 EC2 執行個體。如需如何變更執行個體類型的詳細資訊，請參閱 Amazon EC2 文件中的[變更執行個體類型](#)。

## 其他資源

- [AWS Compute Optimizer 識別和篩選 Microsoft SQL Server 工作負載](#) (AWS)
- [最佳化效能並降低授權成本：利用 AWS Compute Optimizer Amazon EC2 SQL Server 執行個體](#) (AWS 部落格)
- [什麼是 AWS Compute Optimizer ?](#) (AWS 文件)
- [檢視 EC2 執行個體建議](#) (AWS 文件)

## 檢閱 SQL Server 工作負載 Trusted Advisor 的建議

### 概觀

[AWS Trusted Advisor](#) 提供建議，協助您遵循 AWS 最佳實務。透過分析您的用量、組態和支出，Trusted Advisor 提供可行的建議來降低成本、改善系統可用性和效能，或協助消除安全漏洞。本節著重於可協助您降低在中操作 SQL Server 工作負載成本的 Trusted Advisor 檢查 AWS 雲端。

### 成本最佳化建議

Trusted Advisor 提供建議，協助您最佳化 Amazon Elastic Compute Cloud (Amazon EC2) 上的 SQL Server 工作負載。檢查會檢查您的 SQL Server 工作負載，並自動列出需要最佳化的執行個體。操作化 Trusted Advisor 建議可以降低成本並改善組織的安全狀態。

以下是著重於 Microsoft SQL Server 的 Trusted Advisor 檢查：

- Microsoft [SQL Server 過度佈建的 Amazon EC2 執行個體](#) – 此檢查會分析執行 SQL Server 的 Amazon EC2 執行個體，並在執行個體超過 SQL Server 軟體 vCPU 限制時提醒您。例如，具有 SQL Server Standard Edition vCPUs。具有 SQL 伺服器 Web 版的執行個體最多可以使用 32 個 vCPUs。

| 版本  | vCPU 最小值 | vCPU 最大值 |
|-----|----------|----------|
| Web | 4        | 32       |

| 版本         | vCPU 最小值 | vCPU 最大值 |
|------------|----------|----------|
| 標準         | 4        | 48       |
| Enterprise | 4        | 作業系統限制   |

- Microsoft [SQL Server 的 Amazon EC2 執行個體整合](#) – 此檢查會分析您的 Amazon EC2 執行個體，並在執行個體少於 SQL Server 授權數量下限時提醒您。您可以合併較小的 SQL 伺服器執行個體，以協助降低成本。如果您有許多小型、包含授權的 SQL Server 執行個體，請考慮合併。根據 [Microsoft SQL Server 2019 授權指南](#)，SQL Server 每個執行個體至少需要 4 個 vCPU 授權。如果您合併這些資料庫，則可以節省授權成本。您可以根據執行個體上的資料庫數量、資料庫大小上限和資料庫大小總計來做出決策。SQL Server 的 Web、Standard 和 Enterprise 版本支援合併。如需詳細資訊，請參閱 [合併 SQL Server 資料庫](#) (Microsoft 部落格文章)。

AWS 不建議僅在一個伺服器上放置大型生產資料庫。不過，您可以合併用於非生產環境的較小環境，例如用於開發、測試和預備環境。這取決於您目前的 SQL Server 用量；如果您有低用量的資料庫，您可以將合併在一個伺服器上。

## 設定 Trusted Advisor

執行下列動作來評估 SQL Server 重點檢查 Trusted Advisor。

1. 登入 AWS Management Console。
2. 開啟 [AWS Trusted Advisor 主控台](#)。
3. 在導覽窗格中的建議下，選擇成本最佳化。
4. 在成本最佳化檢查清單中，檢閱 Microsoft SQL Server 的 Amazon EC2 執行個體整合狀態，以及針對 Microsoft SQL Server 檢查過度佈建的 Amazon EC2 執行個體狀態。
  - 綠色檢查符號表示您的 Amazon EC2 執行個體已進行最佳設定。
  - 橘色提醒符號表示有改進的機會。
5. 選擇檢查以查看其詳細資訊和建議。
6. 依照檢查提供的指示，最佳化執行 SQL Server 工作負載的 Amazon EC2 執行個體。
7. 定期監控您的執行個體，並定期重新整理檢查。

## 其他資源

- [Trusted Advisor 檢查參考](#) (AWS 文件)

- (AWS) [上的 Microsoft SQL Server AWS](#)
- [上的 Microsoft 授權 AWS](#) (AWS)
- [SQL Server 2019 定價](#) (Microsoft)
- [AWS Launch Wizard for SQL Server](#) (AWS 文件 )

# 容器

現代化是一種轉型的旅程，提供許多選項，包括將整體分解為微服務、使用無伺服器函數 (AWS Lambda) 重新建構以事件驅動的應用程式，以及將資料庫從 SQL Server 重新配置至 Amazon Aurora 或專用受管資料庫。將 .NET Framework 應用程式轉換為 Linux 和 Windows 容器的現代化途徑比其他現代化選項需要更少的精力。容器提供下列優點：

- 加速創新 – 遷移至容器可讓您更輕鬆地自動化開發生命週期的階段，包括建置、測試和部署應用程式。透過自動化這些程序，開發和營運團隊有更多時間專注於創新。
- 降低總擁有成本 (TCO) – 移至容器也可以降低您對授權管理和端點保護工具的依賴。由於容器是暫時性的運算單位，因此您可以自動化和簡化管理任務，例如修補、擴展和備份和還原。這可降低管理和操作容器型工作負載的 TCO。最後，與虛擬機器相比，容器更有效率，因為您可以使用容器，藉由提供更好的隔離來最大化應用程式的置放。這會增加應用程式基礎設施資源的使用率。
- 改善資源使用率 – 與虛擬機器相比，容器更有效率，因為您可以使用容器來最大化應用程式的位置。這可透過提供更好的隔離來提高應用程式的基礎設施資源使用率。
- 縮小技能差距 – 為開發團隊 AWS 提供容器技術和 DevOps 實務方面的技能提升臨場期。

本節涵蓋下列主題：

- [將 Windows 應用程式移至容器](#)
- [最佳化 Amazon ECS 上 AWS Fargate 任務的成本](#)
- [取得 Amazon EKS 成本的可見性](#)
- [使用 App2Container 複寫 Windows 應用程式](#)

如需授權資訊，請參閱 Amazon Web Services 和 Microsoft 的授權一節：常見問答集或將您的問題透過電子郵件傳送至 [microsoft@amazon.com](mailto:microsoft@amazon.com)。 <https://aws.amazon.com/windows/faq/#licensing-q>

## 將 Windows 應用程式移至容器

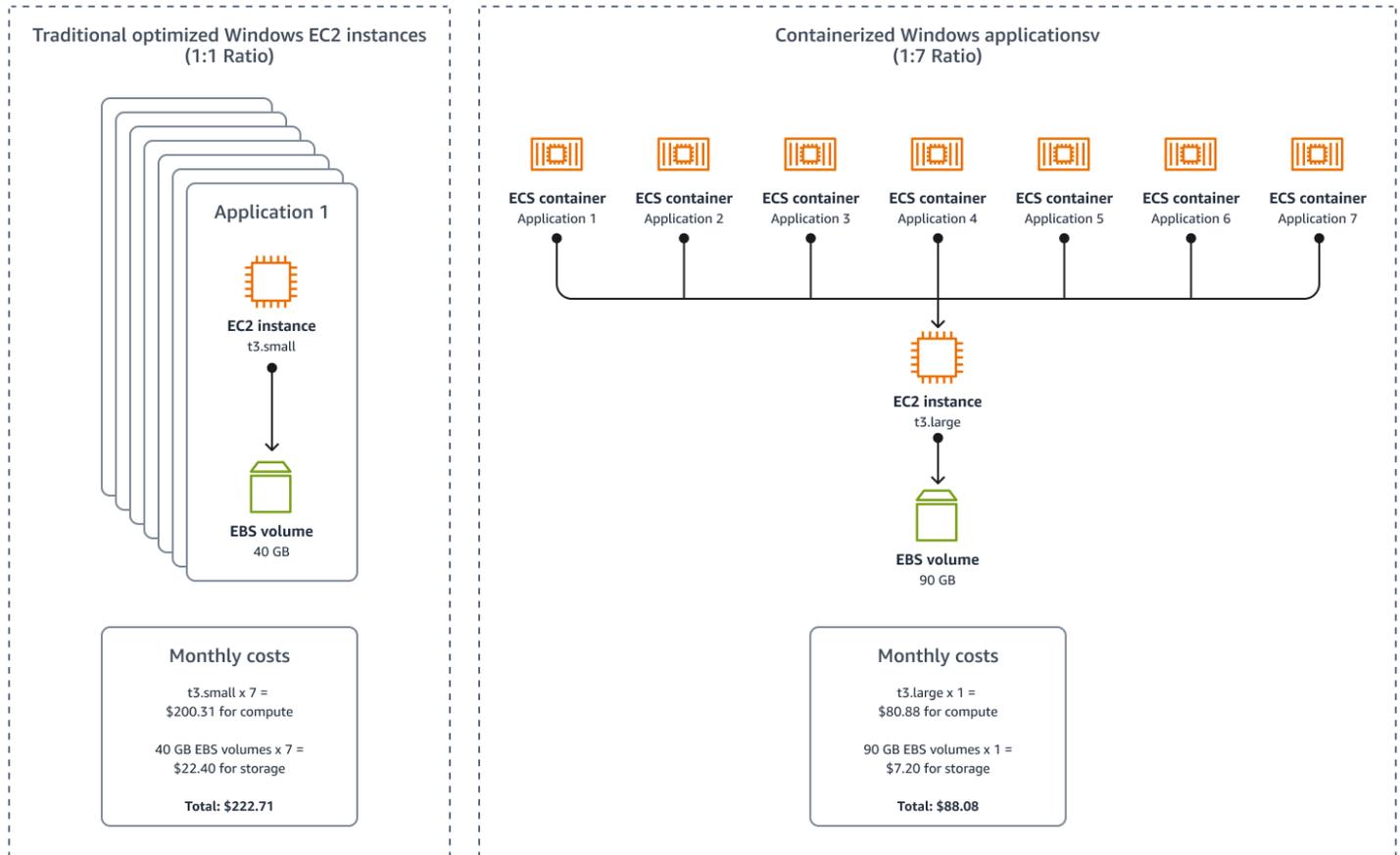
### 概觀

根據 [CNCF 2021 年度調查](#)，96% 的組織正在使用或評估容器來現代化其基礎設施。這是因為容器可協助您的組織降低風險、提高營運效率和速度，以及啟用敏捷性。您也可以使用容器來降低執行應用程式的成本。本節提供跨容器服務以經濟實惠的方式執行 AWS 容器的建議，包括 [Amazon Elastic](#)

## Container Service (Amazon ECS)、Amazon Elastic Kubernetes Service (Amazon EKS) 和 AWS Fargate。

### 成本利益

下列資訊圖表顯示企業可以透過根據[AWS 最佳化和授權評估 \(AWS OLA\)](#) 建議，將其 ASP.NET Framework 應用程式合併到 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體，來節省成本。下列資訊圖表顯示將應用程式移至 Windows 容器可以達成哪些額外節省。



OLA AWS 建議企業進行提升，並轉移到個別的 t3.small 執行個體。如下列效能使用率分析所示，企業可以在內部部署伺服器上執行七個 ASP.NET 應用程式，藉此節省這些成本。

| Server name   | Storage | Operating system    | On-premises CPU AVG utilization | On-premises CPU peak utilization | On-premises RAM (GB) | On-premises RAM AVG utilization (GB) | On-premises RAM peak utilization (GB) | Instance size | vCPU | RAM (GB) |
|---------------|---------|---------------------|---------------------------------|----------------------------------|----------------------|--------------------------------------|---------------------------------------|---------------|------|----------|
| 1 AppServer01 | 60      | Windows Server 2012 | 7.00%                           | 17.00%                           | 8                    | 13.50%                               | 17.10%                                | t3.small      | 2    | 2        |
| 2 AppServer02 | 39      | Windows Server 2012 | 20.07%                          | 22.00%                           | 16                   | 7.50%                                | 12.40%                                | t3.small      | 2    | 2        |
| 3 AppServer03 | 39      | Windows Server 2012 | 24.00%                          | 25.50%                           | 16                   | 8.80%                                | 11.90%                                | t3.small      | 2    | 2        |
| 4 AppServer04 | 4       | Windows Server 2012 | 21.40%                          | 24.00%                           | 16                   | 7.80%                                | 10.70%                                | t3.small      | 2    | 2        |
| 5 AppServer05 | 40      | Windows Server 2012 | 21.30%                          | 23.00%                           | 16                   | 8.20%                                | 12.00%                                | t3.small      | 2    | 2        |
| 6 AppServer06 | 39      | Windows Server 2012 | 21.50%                          | 23.50%                           | 16                   | 7.90%                                | 10.90%                                | t3.small      | 2    | 2        |
| 7 AppServer07 | 39      | Windows Server 2012 | 21.60%                          | 22.90%                           | 16                   | 8.40%                                | 11.50%                                | t3.small      | 2    | 2        |

進一步分析顯示，企業可以在容器上執行工作負載，以節省更多成本。容器可降低 CPU、RAM 和磁碟用量等系統資源的作業系統額外負荷（下一節說明）。在此案例中，企業可以將所有七個應用程式合併到一個 t3.large 執行個體，但仍有 3 GB 的 RAM 可供備用。遷移至容器有助於企業使用容器而非 Amazon EC2，在運算和儲存之間平均節省 64% 的成本。

## 成本最佳化建議

下一節提供透過合併應用程式和使用容器來最佳化成本的建議。

### 減少 Amazon EC2 上的 Windows 使用量

Windows 容器可讓您將更多應用程式合併到較少 Amazon EC2 上的 Windows 佔用空間。例如，假設您有 500 個 ASP.NET 應用程式。如果您在 Amazon EC2 上執行一個 Windows 應用程式的核心，等於 500 個 Windows 執行個體 (t3.small)。如果您假設使用 Windows 容器（使用 t3.large）的比率為 1：7（可能會根據 EC2 執行個體類型/大小而大幅增加），則只需要大約 71 個 Windows 執行個體。這表示您的 Windows on Amazon EC2 足跡減少了 85.8%。

### 降低 Windows 授權成本

如果您授權 Windows 執行個體，則不需要授權在該執行個體上執行的容器。因此，使用 Windows 容器合併 ASP.NET 應用程式可以大幅降低您的 Windows 授權成本。

### 減少您的儲存足跡

每次啟動新的 EC2 執行個體時，您都會建立並支付新的 Amazon Elastic Block Store (Amazon EBS) 磁碟區來存放作業系統。隨著規模擴展，成本也會隨之擴展。如果您使用容器，可以降低儲存成本，因為所有容器共用相同的基礎作業系統。此外，容器會使用 layer 的概念，根據該映像，為所有執行中的容器重複使用容器映像的不可變部分。在上述範例案例中，所有容器都在執行 .NET Framework，因此都共用中繼和不可變的 ASP.NET Framework layer。

## 將end-of-support伺服器遷移至容器

對 Windows Server 2012 和 Windows Server 2012 R2 的支援已於 2023 年 10 月 10 日結束。您可以將在 Windows Server 2012 或舊版上執行的應用程式容器化，以便在新的作業系統上執行，藉此遷移應用程式。如此一來，您就可以避免在不合規的作業系統上執行應用程式，同時利用容器提供的成本效率、降低風險、營運效率、速度和敏捷性。

使用此方法需要注意的事項是，如果您的應用程式需要與目前使用中的作業系統版本相關的特定 APIs (例如 COM Interop)。在此情況下，您必須測試將應用程式移至較新的 Windows 版本。Windows 容器會將基本容器映像 (例如 Windows Server 2019) 與容器主機的作業系統 (例如 Windows Server 2019) 對齊。測試和移至容器，可讓您在未來更輕鬆地進行作業系統升級，方法是變更 Dockerfile 內的基本映像，並部署到執行最新版本 Windows 的新主機集。

## 移除第三方管理工具和授權

管理您的伺服器機群需要使用數種第三方系統操作工具進行修補和組態管理。這些可能會讓基礎設施管理變得複雜，而且您通常會產生第三方授權成本。如果您在上使用容器 AWS，則不需要在作業系統端管理任何內容。容器執行期會管理容器。這表示基礎主機是暫時性的，可以輕鬆取代。您可以執行容器，而無需直接管理容器主機。此外，您可以使用等免費工具 AWS Systems Manager Session Manager，輕鬆存取主機並疑難排解問題。

## 改善控制和可攜性

容器可讓您比透過 EC2 執行個體更精細地控制 CPU 和 RAM 等伺服器資源。對於 EC2 執行個體，您可以選取執行個體系列、執行個體類型和 CPU [選項來控制 CPU](#) 和 RAM。不過，使用容器時，您可以確切定義要配置給 ECS 任務定義中容器或 [Amazon EKS 中 Pod](#) 的 CPU 或 RAM 數量。事實上，我們建議為 Windows [容器指定容器層級的 CPU 和記憶體](#)。此精細程度可帶來成本效益。請考慮下列範例程式碼：

```
json
{
  "taskDefinitionArn": "arn:aws:ecs:us-east-1:123456789012:task-definition/demo-service:1",
  "containerDefinitions": [
    {
      "name": "demo-service",
      "image": "mcr.microsoft.com/dotnet/framework/samples:aspnetapp-windowsservercore-ltsc2019",
      "cpu": 512,
      "memory": 512,
      "links": [],
```

```
"portMappings": [  
  {  
    "containerPort": 80,  
    "hostPort": 0,  
    "protocol": "tcp"  
  }  
],
```

## 加速創新

移至容器可讓您更輕鬆地自動化開發生命週期的階段，包括建置、測試和部署應用程式。如果您將這些程序自動化，您可以為您的開發和營運團隊提供更多時間專注於創新。

## 降低 TCO

移至容器通常會降低對授權管理和端點保護工具的依賴。由於容器是暫時性的運算單位，因此您可以自動化和簡化管理任務，例如修補、擴展和備份和還原。這可以降低管理和操作容器型工作負載的 TCO。與虛擬機器相比，容器更有效率，因為它們可讓您最大化應用程式的置放，以便您可以提高應用程式的基礎設施資源使用率。

## 關閉技能差距

AWS 提供計畫和臨場日，以提升容器和 DevOps 技術的客戶開發團隊技能。這包括實作諮詢和啟用。

## 重構 .NET 5+ 並使用 Linux 容器

雖然您可以將 .NET Framework 應用程式移至容器來降低成本，但當您將舊版 .NET 應用程式重構為雲端原生替代方案時，可以進一步節省成本 AWS。

## 移除授權成本

將您的應用程式從 Windows 上的 .NET Framework 重構為 Linux 上的 .NET Core 可節省約 45% 的成本。

## 存取最新的增強功能

將您的應用程式從 Windows 上的 .NET Framework 重構為 Linux 上的 .NET Core，可讓您存取最新的增強功能，例如 Graviton2。Graviton2 比同級執行個體提供 40% 更好的效能價格。

## 改善安全性和效能

將您的應用程式從 Windows 上的 .NET Framework 重構為 Linux 容器上的 .NET Core，可改善安全性和效能。這是因為您取得最新的安全修補程式、從容器隔離中受益，以及可存取新功能。

## 使用 Windows 容器，而不是在一個 IIS 執行個體上執行許多應用程式

考慮使用 Windows 容器，而不是在具有網際網路資訊服務 (IIS) 的 EC2 Windows 執行個體上執行多個應用程式的優勢：

- 安全 – 容器提供開箱即用的層級安全，而此層級並非透過 IIS 層級的隔離達成。如果一個 IIS 網站或應用程式遭到入侵，所有其他託管網站都會公開且易受攻擊。容器逸出很少見，而且與透過 Web 漏洞取得伺服器控制權相比，更難利用。
- 彈性 – 能夠在程序隔離中執行容器，並擁有自己的執行個體，允許更精細的網路選項。容器也在許多 EC2 執行個體中提供複雜的分佈方法。當您在單一 IIS 執行個體上合併應用程式時，不會獲得這些好處。
- 管理開銷 – 伺服器名稱指示 (SNI) 會建立需要管理和自動化的開銷。此外，您必須掌握典型的作業系統管理操作，例如修補、故障診斷 BSOD（如果沒有自動擴展）、端點保護等。根據[安全最佳實務](#)設定 IIS 網站是一項耗時且持續的活動。您甚至可能需要設定[信任層級](#)，這也會增加管理開銷。容器設計為無狀態且不變。最後，如果您改用 Windows 容器，您的部署會更快、更安全且可重複。

## 後續步驟

投資現代基礎設施來執行舊版工作負載，為您的組織帶來巨大的好處。AWS 容器服務可讓您更輕鬆地管理基礎基礎設施，無論是在內部部署還是雲端，因此您可以專注於創新和業務需求。雲端中將近 80% 的容器 AWS 現在都會在上執行。為幾乎所有使用案例 AWS 提供豐富的容器服務。若要開始使用，請參閱[位於的容器 AWS](#)。

## 其他資源

- [使用 ECS 容量提供者和 EC2 Spot 執行個體來最佳化容器工作負載的成本](#) (AWS 部落格)
- [Amazon ECS 和 \( 部落格 \) 的成本最佳化檢查清單 AWS Fargate](#) AWS
- [Amazon EKS on AWS Graviton2 全面推出：多架構應用程式的考量事項](#) (AWS 部落格)
- [上的 Kubernetes 成本最佳化 AWS](#) (AWS 部落格)
- [使用 Karpenter 合併來最佳化 Kubernetes 運算成本](#) (AWS 部落格)

# 最佳化 Amazon ECS 上 AWS Fargate 任務的成本

## 概觀

適當的規模調整 AWS Fargate 任務是成本最佳化的重要步驟。很多時候，應用程式是使用 Fargate 任務的任意大小來建置，而且永遠不會重新檢視。這可能會導致 Fargate 任務過度佈建和不必要的花費。本節說明如何使用 [AWS Compute Optimizer](#) 提供可行的建議，以便您可以最佳化在 Fargate 上執行的 Amazon Elastic Container Service (Amazon ECS) 服務的任務 CPU 和記憶體。Compute Optimizer 也會量化採用這些建議的成本影響。這可讓您根據節省機會的大小來排定最佳化工作的優先順序。Compute Optimizer 建議提供容器層級 CPU 和記憶體組態，以縮減任務大小。

## 成本利益

在 Fargate 上正確調整 Amazon ECS 任務的大小，可以為長時間執行的任務降低成本 30–70%。在不檢閱應用程式效能指標以正確調整任務大小的情況下，您可以將 EC2 運算執行個體上使用的相同思維套用至容器大小。這會導致 Fargate 任務過大，進而增加閒置資源的成本。您可以使用 Compute Optimizer，以反應方式呈現正確的大小調整機會。理想情況下，應用程式擁有人會檢閱特定應用程式效能指標，並移除作業系統額外負荷，以確保指定適當的任務大小。如需詳細資訊，請參閱本指南的將 [Windows 應用程式移至容器](#) 一節。

## 成本最佳化建議

本節提供使用 Compute Optimizer 在 Fargate 任務上正確調整 Amazon ECS 大小的建議。

作為成本最佳化程序的一部分，我們建議您執行下列動作：

- 啟用運算最佳化工具
- 使用 Compute Optimizer 結果
- 標記要調整大小的任務
- 啟用成本分配標籤以使用 AWS 帳單工具
- 實作正確的大小調整建議
- 在 Cost Explorer 中檢閱成本前後

## 啟用運算最佳化工具

您可以在 [AWS Compute Optimizer](#) 中的組織或單一帳戶層級啟用 AWS Organizations。整個組織的組態會針對所有成員帳戶的整個機群中新的和現有的執行個體提供持續報告。這可讓正確的大小調整為重複活動，而不是 point-in-time 活動。

## 組織層級

對於大多數組織而言，使用 Compute Optimizer 最有效率的方式是在組織層級。這可提供多帳戶和多區域的組織可見性，並將資料集中到一個來源以供檢閱。若要在組織層級啟用此功能，請執行下列動作：

1. 使用具有**必要許可**的角色登入您的[AWS Organizations 管理帳戶](#)，並選擇加入此組織中的所有帳戶。您的組織必須**啟用所有功能**。
2. 啟用管理帳戶後，您可以登入帳戶、查看所有其他成員帳戶，並瀏覽他們的建議。

### Note

最佳實務是設定 Compute Optimizer 的[委派管理員帳戶](#)。這可讓您執行最低權限原則，將 AWS Organizations 管理帳戶的存取權降至最低，同時仍提供整個組織的服務存取權。

## 單一帳戶層級

如果您以成本高的帳戶為目標，但無法存取 AWS Organizations，您仍然可以為該帳戶和區域啟用 Compute Optimizer。若要了解選擇加入程序，請參閱 [入門 AWS Compute Optimizer](#)。

### Note

建議會每天重新整理，最多可能需要 12 小時才能產生。請記住，Compute Optimizer 在過去 14 天內需要 24 小時的指標，才能為 Fargate 上的 Amazon ECS 產生建議。如需詳細資訊，請參閱 Compute Optimizer 文件中的 [Fargate 上的 Amazon ECS 服務需求](#)。

Compute Optimizer 會自動分析 Fargate 上 Amazon ECS 服務的下列 Amazon CloudWatch 和 Amazon ECS 使用率指標：

- CPUUtilization – 服務中使用的 CPU 容量百分比。
- MemoryUtilization – 服務中使用的記憶體百分比。

## 使用 Compute Optimizer 結果

請考慮一個範例，專注於在單一帳戶和單一區域中進行正確的大小調整。在此範例中，運算最佳化工具會在所有帳戶的組織層級啟用。請記住，正確調整大小是一種破壞性程序，在大多數情況下，應用程式擁有人會在數週的排程維護時段中，以精確度執行。

如果您在組織的管理帳戶中導覽至 Compute Optimizer（如下列步驟所示），您可以選擇要調查的帳戶。在此範例中，一個任務正在中過度佈建的單一帳戶中執行 us-east-1。重點是調整 Amazon ECS 服務的建議大小。

1. 開啟 [Compute Optimizer 主控台](#)。
2. 在儀表板頁面上，依問題清單篩選=佈建過高，以查看 Fargate 上的所有 Amazon ECS 服務。
3. 若要檢閱 Fargate 上過度佈建 ECS 服務的詳細建議，請向下捲動，然後選擇檢視建議。
4. 選擇匯出並儲存檔案以供日後使用。

### Note

若要儲存建議以供未來檢閱，您必須有一個 S3 儲存貯體可供 Compute Optimizer 在每個區域中寫入。如需詳細資訊，請參閱 Compute Optimizer 文件中的 [適用於的 Amazon S3 儲存貯體政策 AWS Compute Optimizer](#)。

若要查看 Compute Optimizer 的建議，請執行下列動作：

1. 在 [Compute Optimizer 主控台](#) 中，前往匯出建議頁面。
2. 針對 S3 儲存貯體目的地，選擇您的 S3 儲存貯體。
3. 在匯出篩選條件區段中，針對資源類型，選擇 Fargate 上的 ECS 服務。
4. 在 Fargate 上 ECS 服務的建議頁面上，深入探索 Fargate 上的其中一個 ECS 服務，並查看 Compute Optimizer 的 CPU 和記憶體建議。例如，檢閱比較目前設定與建議任務大小的建議，以及比較目前設定與建議容器大小區段的建議。

若要取得適當大小所需的 Fargate ECS 服務清單，請執行下列動作：

1. 開啟 [Amazon S3 主控台](#)。
2. 在導覽窗格中，選擇儲存貯體，然後選擇您匯出結果的儲存貯體。
3. 在物件索引標籤上，選取您的物件，然後選擇下載。

4. 在您下載的結果中，篩選問題清單欄，以在 Fargate 上僅顯示 OVER\_PROVISIONED Amazon ECS 服務。這會顯示您計劃針對適當大小設定目標的 Amazon ECS 服務。
5. 將任務定義存放在文字編輯器中以供日後使用。

## 正確調整標籤任務的大小

標記工作負載是強大的工具，可在中組織您的資源 AWS。您可以使用標籤來深入了解成本並啟用退款。有許多方法和策略可將標籤新增至資源，AWS 以處理扣款和自動化。如需詳細資訊，請參閱白皮書標記資源的 AWS 最佳實務。[AWS](#) 下列範例使用 [AWS CloudShell](#) 來標記屬於目標帳戶和 中任何 Amazon ECS 服務一部分的所有任務 AWS 區域。

```
#!/bin/bash
# Set variables
TAG_KEY="rightsizing"
TAG_VALUE="enabled"
# Get a list of ECS Clusters
ClustersArns=$( w secs list-clusters -query 'clusterArns' -output text)
for ClustersArn in $ClustersArns; do
  ServiceArns=$( w secs list-services -cluster $ClustersArn -query 'serviceArns' -output text)
  for ServiceArn in $ServiceArns; do
    TasksArns=$( w secs list-tasks -cluster $ClustersArn -service-name $ServiceArn -query 'taskArns' -output text)
    for TasksArn in $TasksArns; do
      w secs tag-resource -resource-arn $TasksArn -tags key=$TAG_KEY,value=$TAG_VALUE
    done
  done
done
```

下列程式碼範例示範如何啟用[標籤傳播](#)至所有 Amazon ECS 服務。

```
#!/bin/bash
# Set variables
TAG_KEY="rightsizing"
TAG_VALUE="enabled"
# Get a list of ECS Clusters
ClustersArns=$(aws ecs list-clusters --query 'clusterArns' --output text)
for ClustersArn in $ClustersArns; do
  ServiceArns=$(aws ecs list-services --cluster $ClustersArn --query 'serviceArns' --output text)
  for ServiceArn in $ServiceArns; do
```

```
aws ecs update-service --cluster $ClustersArn --service $ServiceArn --propagate-tags SERVICE &>/dev/null
aws ecs tag-resource --resource-arn $ServiceArn --tags key=$TAG_KEY,value=$TAG_VALUE
done
done
```

## 啟用成本分配標籤以使用 AWS 帳單工具

我們建議您啟用使用者定義的成本分配標籤。這可讓 AWS 計費工具（例如 AWS Cost Explorer 和）中識別和篩選 Rightsizing 標籤 AWS Cost and Usage Report。如果您未啟用此功能，則標籤篩選選項和資料將無法使用。如需使用成本分配標籤的資訊，請參閱 文件中的 AWS 帳單與成本管理 [啟用使用者定義的成本分配標籤](#)。

等待 24 小時後，您可以在 Cost Explorer 中看到標籤，然後在下一節實作正確的大小調整建議。若要執行此操作，請在 Cost Explorer 中搜尋 Rightsizing 標籤。

## 實作正確的大小調整建議

Compute Optimizer 將提供任務或容器大小建議。若要實作正確的大小調整建議，請執行下列動作。

1. 開啟 [Amazon ECS 主控台](#)。
2. 從導覽列中選擇包含您任務定義的區域。
3. 在導覽窗格中，選擇 Task Definitions (任務定義)。
4. 在 Task definitions (任務定義) 頁面上，選擇任務，然後選擇 Create new revision (建立新修訂版)。
5. 在 Create new task definition revision (建立新任務定義修訂版) 頁面上進行變更。若要更新容器大小建議，請在 [ECS 任務定義](#) 中的 containerDefinitions 區塊 memory 下更新 cpu 和 。例如：

```
"containerDefinitions": [
  {
    "name": "your-container-name",
    "image": "your-image",
    "cpu": 1024,
    "memory": 2048,
  }
],
```

6. 驗證資訊，然後選擇 Create (建立)。

若要更新 Amazon ECS 服務，請執行下列動作：

1. 開啟 [Amazon ECS 主控台](#)。
2. 在 Clusters (叢集) 頁面上，選取您的叢集。
3. 在 Cluster overview (叢集概觀) 頁面中，選取服務，然後選擇 Update (更新)。
4. 針對 Task definition (任務定義)，選擇要使用的任務定義系列和修訂。

對於進階運算子，您可以使用 CloudShell 來更新 Amazon ECS 服務。例如：

```
bash
#!/bin/bash
# Set variables
ClustersName="workshop-cluster"
ServiceName="lab7-fargate-service"
TaskDefinition="lab7-fargate-demo:3"
# update the service
aws ecs update-service --cluster $ClustersName --service $ServiceName --task-definition
$TaskDefinition
```

## 檢閱成本前後

在您調整資源大小後，您可以使用 Cost Explorer，使用 Rightsizing 標籤在成本前後顯示。請記住，您可以使用 [資源標籤](#) 來追蹤成本。透過使用多層標籤，您可以精細地了解成本。在本指南涵蓋的範例中，Rightizing 標籤用於將一般標籤套用至所有目標執行個體。然後，團隊標籤會用來進一步組織資源。下一個步驟是介紹應用程式標籤，以進一步顯示操作特定應用程式的成本影響。

考慮一個範例，說明在單一帳戶層級使用 Rightsizing 標籤可以達成的成本降低。在此範例中，營運成本從每天 30.26 美元降至每天 7.56 美元。假設每月 744 小時，正確調整規模之前的年度成本為 11,044.9 美元。在正確調整規模之後，年成本會降至 2,759.4 美元。這表示此帳戶的運算成本降低了 75%。想像一下，這在大型組織中的影響。

開始正確的規模調整旅程之前，請考慮下列事項：

- AWS 提供許多降低成本的選項。這包括 [AWS OLA](#)，其中會先 AWS 檢閱您的內部部署執行個體，再移至 AWS。OLA AWS 也為您提供適當大小的建議和授權指導。
- 購買 [Savings Plans](#) 之前，請先完成所有正確的大小調整。這可協助您避免在 Savings Plans 承諾上過度購買。

## 後續步驟

我們建議執行下列步驟：

1. 檢閱您現有的環境，並考慮將 Amazon EBS gp2 磁碟區轉換為 gp3 磁碟區。
2. 檢閱 [Savings Plans](#)。

## 其他資源

- [Compute Optimizer 入門](#) (AWS 文件)
- [標記 AWS 資源的最佳實務](#) (AWS 白皮書)
- [Windows 容器位於 AWS](#)(AWS Workshop Studio)

## 取得 Amazon EKS 成本的可見性

### 概觀

為了有效監控 Kubernetes 部署的成本，全方位檢視是必要的。唯一的固定和已知成本是 Amazon Elastic Kubernetes Service (Amazon EKS) 控制平面。這包括組成部署的所有其他元件，從運算和儲存到聯網，都是根據您的應用程式需求的變數數量。

您可以使用 [Kubecost](#) 分析 Kubernetes 基礎設施的成本，從[命名空間和服務](#)一直到個別 [Pod](#)，然後在儀表中顯示資料。Kubecost 表面叢集內成本，例如運算和儲存，以及 out-of-cluster 成本。[Amazon S3](#) [Amazon Relational Database Service](#) Kubecost 會根據此資料提出適當大小的建議，並顯示可能影響系統的關鍵警示。Kubecost 可與 [整合 AWS Cost and Usage Report](#)，以顯示 [Compute Savings Plans](#)、[預留執行個體](#)和其他折扣計劃的節省。

### 成本利益

Kubecost 提供報告和儀表板，可視覺化 Amazon EKS 部署的成本。它可讓您從叢集向下切入各個元件，例如控制器、服務、節點、Pod 和磁碟區。這可讓您全面檢視在 Amazon EKS 環境中執行的應用程式。透過啟用此可見性，您可以根據 Kubecost 建議採取行動，或精細檢視每個應用程式的成本。正確調整 Amazon EKS 節點群組的大小可提供與標準 EC2 執行個體相同的潛在節省。如果您可以正確調整容器和節點的大小，則可以從執行容器所需的執行個體大小，以及自動擴展群組中所需的 EC2 執行個體數量中移除運算浮水印。

## 成本最佳化建議

若要利用 Kubecost，建議您執行下列動作：

1. 將 Kubecost 部署到您的環境
2. 取得 Windows 應用程式的精細成本明細
3. 正確大小的叢集節點
4. 正確大小的容器請求
5. 管理未充分利用的節點
6. 修正已捨棄的工作負載
7. 根據建議採取行動
8. 更新自我管理節點

### 將 Kubecost 部署到您的環境

[Amazon EKS Finhack 研討會](#)會教導您如何在 AWS 擁有的帳戶中部署設定為使用 Kubecost 的 Amazon EKS 環境。這可讓您取得技術的實作體驗。如果您有興趣在組織中執行此研討會，請聯絡您的客戶團隊。

若要使用 [Helm](#) 將 Kubecost 部署到您的 Amazon EKS 叢集，請參閱 [AWS 和 Kubecost 協作，為部落格上的 EKS 客戶提供成本監控](#)。AWS 或者，您可以參閱[官方 Kubecost 文件](#)，以取得安裝和設定 Kubecost 的說明。如需有關 Kubecost 支援 Windows 節點的資訊，請參閱 [Kubecost 文件中的 Windows Node Support](#)。

### 取得 Windows 應用程式的精細成本明細

雖然您可以使用 [Amazon EC2 Spot 執行個體](#)來大幅節省成本，但您也可以受益於 Windows 工作負載通常具有狀態。Spot 執行個體的使用取決於應用程式，建議您驗證它們是否適用於您的使用案例。

若要取得 Windows 應用程式的精細成本明細，[請登入 Kubecost](#)。在導覽頁面中，選擇節省。

### 正確大小的叢集節點

在 [Kubecost](#) 中，從導覽列選擇 Savings，然後選擇正確大小的叢集節點。

請考慮一個範例，其中 Kubecost 報告叢集在 vCPU 和 RAM 方面都過度佈建。下表顯示 Kubecost 的詳細資訊和建議。

|        | Current             | 建議：簡單        | 建議：複雜                     |
|--------|---------------------|--------------|---------------------------|
| 總計數    | 每月 3462.57 美元       | 每月 137.24 美元 | 每月 303.68 美元              |
| 節點計數   | 4                   | 5            | 4                         |
| CPU    | 74 VCPUs            | 10 VCPUs     | 8 VCPUs                   |
| RAM    | 152 GB              | 20 GB        | 18 GB                     |
| 執行個體明細 | 2 c5.xlarge + 2 個其他 | 5 t3a.medium | 2 個 c5n.large + 其他<br>1 個 |

如 Kubecost 部落格文章所述 [尋找 Kubernetes 叢集的最佳節點集](#)，簡單選項會使用單一節點群組，而複雜節點則使用多節點群組方法。了解如何採用按鈕可執行一鍵式叢集調整大小。它需要安裝 [Kubecost 叢集控制器](#)。

如果您使用的是非由 [eksctl](#) 建立的 [自我管理 Windows 節點](#)，請參閱 [更新現有的自我管理節點群組](#)。這些指示說明如何在 [Auto Scaling 群組](#) 使用的 Amazon EC2 啟動範本中變更執行個體類型。

## 正確大小的容器請求

在 [Kubecost](#) 中，從導覽列選擇 Savings，然後前往請求正確大小建議頁面。此頁面顯示 Pod 的 [效率](#)、適當大小的建議，以及預估的成本節省。您可以使用自訂按鈕，依叢集、節點、命名空間\控制器等進行篩選。

例如，假設 Kubecost 已計算出您的一些 Pod 在 CPU 和 RAM（記憶體）方面過度佈建。然後，Kubecost 建議您調整為新的 CPU 和 RAM 值，以實現其估計每月節省。若要變更 CPU 和 RAM 值，您必須更新 [部署資訊清單](#) 檔案。

## 管理未充分利用的節點

在 [Kubecost](#) 中，從導覽列選擇 Savings，然後選擇管理未充分利用的節點。

請考慮一個範例，其中頁面顯示叢集中的一個節點在 CPU 和 RAM（記憶體）方面未充分利用，因此可以耗盡，並終止或調整大小。選擇未通過節點和 Pod 檢查的節點，將為您提供無法耗盡節點的原因的詳細資訊。

## Remedy 已捨棄的工作負載

在 [Kubecost](#) 中，從導覽列選擇 Savings，然後選擇放棄工作負載頁面。在此範例中，您會依名為 windows 的命名空間進行篩選。此頁面顯示不符合流量閾值且被視為已放棄的 Pod。Pod 需要在定義的期間內傳送或接收特定數量的網路流量。

仔細考慮一個或多個 Pod 已捨棄後，您可以縮減複本數量、刪除部署、調整其大小以使用較少的資源，或通知應用程式擁有者您認為已捨棄部署，以節省成本。

### 根據建議採取行動

在適當大小的叢集節點區段中，Kubecost 會分析叢集中工作者節點的使用情形，並針對正確調整節點大小以降低成本提出建議。有兩種類型的節點群組可與 Amazon EKS 搭配使用：[自我管理和受管](#)。

### 更新自我管理節點

如需有關更新自我管理節點的資訊，請參閱 Amazon EKS 文件中的[自我管理節點更新](#)。其中指出 eksctl 無法使用 建立的節點群組無法更新，且必須遷移至具有新組態的新節點群組。

例如，假設您有一個名為 ng-windows-m5-2xlarge（使用 m5.2xlarge EC2 執行個體）的 Windows 節點群組，而且您想要將 Pod 遷移到名為 ng-windows-t3-large（由 t3.large EC2 執行個體支援以節省成本）[的新節點群組](#)。

若要在使用 所部署的節點群組時遷移至新的節點群組 eksctl，請執行下列動作：

1. 若要尋找 Pod 目前的節點，請執行 `kubectl describe pod <pod_name> -n <namespace>` 命令。
2. 執行 `kubectl describe node <node_name>` 命令。輸出顯示節點正在 m5.2xlarge 執行個體上執行。它也符合節點群組名稱 (ng-windows-m5-2xlarge)。
3. 若要變更部署以使用節點群組 ng-windows-t3-large，請刪除節點群組 ng-windows-m5-2xlarge 並執行 `kubectl describe svc,deploy,pod -n windows`。現在已刪除其節點群組後，部署立即開始重新部署。

#### Note

當您刪除節點群組時，服務將會停機。

4. 幾分鐘後再次執行 `kubectl describe svc,deploy,pod -n windows` 命令。輸出顯示 Pod 再次處於執行中狀態。

5. 若要顯示 Pod 現在正在節點群組 上執行ng-windows-t3-large，請再次執行 `kubectl describe pod <pod_name> -n <namespace>`和 `kubectl describe node <node_name>`命令。

## 替代的調整大小方法

此方法適用於自我管理或受管節點群組的任何組合。[無縫遷移工作負載從 EKS 自我管理節點群組到 EKS 受管節點群組](#) 部落格文章提供指引，說明如何將工作負載從具有超大執行個體類型的一個節點群組遷移到大小正確的節點群組，而不會停機。

## 後續步驟

Kubecost 可讓您輕鬆地將 Amazon EKS 環境的成本視覺化。Kubecost 與 Kubernetes 和 AWS APIs 的深度整合可協助您找到潛在的成本節省。您可以在 Kubecost 的 Savings 儀表板中將這些視為建議。Kubecost 也可以透過其[叢集控制器功能](#)，為您實作其中一些建議。

我們建議您檢閱 [和 Kubecost 協作中的step-by-step部署](#)，以從 AWS 容器部落格為 EKS 客戶部落格文章提供成本監控。[AWS](#)

## 其他資源

- [Amazon EKS 研討會](#) (Amazon EKS 研討會 )
- [AWS 和 Kubecost 合作為 EKS 客戶提供成本監控](#) (AWS 部落格 )
- [Amazon EKS Finhack 研討會](#) (AWS Workshop Studio)
- [Windows Containers on AWS](#)(AWS Workshop Studio)

## 使用 App2Container 複寫 Windows 應用程式

### 概觀

[AWS App2Container](#) 是一種命令列工具，可將 Java 和 .NET Web 應用程式遷移和現代化為容器。App2Container 會分析並建置在裸機、虛擬機器、Amazon Elastic Compute Cloud (Amazon EC2) 執行個體或其他雲端提供者中執行的所有應用程式的清查。您選取要容器化的應用程式。App2Container 將應用程式成品和相依性封裝到容器映像中、設定網路連接埠，並產生必要的 Amazon Elastic Container Service (Amazon ECS) 和 Amazon Elastic Kubernetes Service (Amazon EKS) 部署成品，這些成品是基礎設施做為程式碼 (IaC) 範本。App2Container 佈建將容器化應用程式

部署到生產環境所需的雲端基礎設施和 CI/CD 管道。如需詳細資訊，請參閱 [App2Container 文件中的 App2Container 運作方式](#)。App2Container

使用 App2Container，您可以將應用程式遷移至 AWS 並現代化為容器，同時標準化應用程式的部署和操作。您可以使用 App2Container 來協助快速建置概念驗證 (PoC)，或加速在容器中部署生產工作負載。

使用 Windows 應用程式時，有幾件事需要記住。App2Container 支援在 Microsoft Internet Information Services (IIS) 上部署的 ASP.NET 應用程式容器化，包括在 Windows Server 2016、Windows Server 2019 或 Windows Server Core 2004 上執行的 IIS 託管 Windows Communication Foundation (WCF) 應用程式。如需詳細資訊，請參閱 App2Container 文件中的 [Windows 支援的應用程式](#)。App2Container 使用 Windows Server Core 做為容器成品的基礎映像，將 Windows Server Core 容器版本與您執行容器化命令的伺服器作業系統 (OS) 版本相符。此方法會將應用程式與基礎作業系統分離，因此您可以升級作業系統，而無需執行傳統遷移。

如果您使用工作者機器將應用程式容器化，容器基礎映像，例如 Windows Server 2019 長期服務管道 (LTSC)，會與您的工作者機器作業系統相符，例如 Windows Server 2019。如果您直接在應用程式伺服器上執行容器化，版本會與您的應用程式伺服器作業系統相符。如果您的應用程式在 Windows Server 2008 或 2012 R2 上執行，您仍然可以透過設定工作者機器進行容器化和部署步驟來使用 App2Container。App2Container 不支援在 Windows 用戶端作業系統上執行的應用程式，例如 Windows 7 或 Windows 10。App2Container 支援適用於 Java 程序的 Tomcat、TomEE 和 JBoss (獨立模式) 架構。如需詳細資訊，請參閱 [App2Container 相容性](#)。

## 成本利益

與 one-application-to-one-server 部署設計模式相比，容器化和整合應用程式可以 [節省高達 60% 的運算成本](#)。App2Container 可協助加速應用程式容器化程序。以下是使用 App2Container 滿足您的現代化需求的一些優點：

- App2Container 免費提供。
- App2Container 支援容器映像中的多個應用程式。
- 使用 App2Container 將舊版 .NET 應用程式移至容器，以解決即將終止支援的作業系統。您可以移至較新的作業系統、避免支付延伸支援的費用，並降低安全風險。
- 容器是封裝 .NET 應用程式的有效且符合成本效益的方法。在 [MACO 建議 - 移至](#) 容器中檢閱容器的優點。
- 應用程式整合和容器化可協助您更有效率地使用運算資源，進而減少運算、儲存和授權足跡。
- 移至容器可以降低營運開銷和基礎設施成本，並提高開發可攜性和部署敏捷性。

## 成本最佳化建議

如需如何使用 App2Container 的說明，請參閱 [入門 AWS App2Container](#)。如需 App2Container 命令的相關資訊，請參閱 [App2Container 命令參考](#)。

## 後續步驟

App2Container 可以加速容器化應用程式和部署至 Amazon EKS 或 Amazon ECS 的程序。將應用程式部署到容器可降低運算、聯網和儲存成本，並降低應用程式運算子的操作開銷。

如需 App2Container 的實際操作體驗，請參閱 [使用 現代化 AWS App2Container 研討會](#)。如果您想要有深入學習體驗，請要求您的 AWS 客戶團隊設定 App2Container 沉浸日。

## 其他資源

- [使用 容器化複雜的多層 Windows 應用程式 AWS App2Container](#) (AWS 部落格文章 )
- [使用 容器化舊版 ASP.NET 應用程式 AWS App2Container](#) (AWS 部落格文章 )
- [App2Container 支援的應用程式](#) (AWS 文件 )
- [使用 AWS App2Container 研討會現代化](#) (AWS 研討會工作室 )
- [AWS App2Container FAQs](#) (AWS 網站 )

# 儲存

為您的 Microsoft 工作負載選擇正確的儲存體是關鍵的架構決策。在決策過程中，我們建議您制定儲存計劃並確定應用程式和服務的功能需求。本章提供下列儲存選項的概觀，這些選項可能會納入您的規劃。

章節：

- [Amazon EBS](#)
- [Amazon FSx](#)
- [AWS Storage Gateway](#)

## Amazon EBS

Amazon Elastic Block Store (Amazon EBS) 是一種全受管的區塊儲存服務，可讓您存放持久性區塊層級儲存磁碟區，以便與 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體搭配使用。您可以利用 Amazon EBS 中的多種功能，有效管理和最佳化雲端中 Windows 工作負載的儲存資源。例如，您可以使用 Amazon EBS 佈建工作負載所需的 IOPS 和輸送量的確切數量、從各種磁碟區類型中選取以符合工作負載需求，並使用工具來識別和消除浪費的儲存資源。這種對儲存效能和用量的精細控制可協助您最佳化儲存資源，同時避免不必要的成本。

本節涵蓋下列主題：

- [將 Amazon EBS 磁碟區從 gp2 遷移至 gp3](#)
- [修改 Amazon EBS 快照](#)
- [刪除未連接的 Amazon EBS 磁碟區](#)

## 將 Amazon EBS 磁碟區從 gp2 遷移至 gp3

### 概觀

固態硬碟 (SSD) 是生產和高效能工作負載的標準儲存選項。Amazon EBS 為中高效能工作負載提供一般用途 SSD 磁碟區。許多 AWS 服務（包括 Amazon EC2）的標準是 [gp2](#)，這是這些一般用途 SSD 磁碟區的第二代。第三代一般用途 SSDs 稱為 [gp3](#)，已於 2020 年 12 月發行。

相較於上一代，gp3 產品在效能自訂方面進行了大幅改善。對於 Amazon EBS gp2 磁碟區，效能會與磁碟區的大小緊密結合。對於每 1 GB 的容量，gp2 磁碟區可獲得 3 IOPS 的效能。也就是說，2,000

GB gp2 磁碟區能夠達到 6,000 IOPS。對於 gp3 磁碟區，可以獨立於儲存容量自訂效能。這可讓小型容量磁碟區達到高達 16,000 IOPS 的效能和每秒 1,000 Mb 的輸送量。

gp3 磁碟區的另一個主要變更是基準 IOPS 效能。gp3 磁碟區從 3,000 IOPS 開始。相較之下，gp2 磁碟區的大小必須達到 1TiB，才能達到相同的效能功能。對於通常具有 C：磁碟機遠小於 1TiB 的 Windows Server，從 gp2 升級到 gp3 是顯著的效能改善。

最後，與 gp2 磁碟區相比，gp3 磁碟區的價格是最大的改善之一。gp3 磁碟區提供增強的效能功能，成本比 gp2 磁碟區低 20%。

## 成本影響

由於能夠獨立於容量擴展效能，請務必了解新增額外 IOPS 和輸送量的定價層面。對於 gp2 磁碟區，定價是以佈建容量為基礎，每 GiB 月 0.10 美元。對於 gp3 磁碟區，定價與高效能[佈建 IOPS SSD 磁碟區](#)類似，其具有一個容量成本，另外還有一個額外的 IOPS 和輸送量成本。

如下表所述，gp3 磁碟區的容量價格為每 GiB 月 0.08 美元（比 gp2 低 20%），而 IOPS 的個別成本為每佈建 IOPS 月 3,000 美元，每佈建 MiBs 每月 0.04 美元 MiBs，以取得輸送量。

|             | gp3   | gp2  |
|-------------|---|--|
| 磁碟區大小       | 1 GiB – 16 TiB  | 1 GiB – 16 TiB   |
| 基準 IOPS     | 3,000   | 3 IOPS/GiB (最低 100 IOPS)，最高 16,000 IOPS<br>小於 1 TiB 的磁碟區最多可爆量 3,000 IOPS |
| 最大 IOPS/磁碟區 | 16,000  | 16,000   |
| 基準輸送量       | 125 MiBs  | 傳輸量限制介於 128 MiBs 250 MiBs之間，取決於磁碟區大小                                     |
| 最大輸送量/磁碟區   | 1,000 MiBs  | 250 MiBs   |
| 價格          | 每月 0.08 USD GiB<br><br>3,000 個 IOPS 免費和 0.005 美元/每月佈建 IOPS，超過 3,000 個 | 0.10 美元/GiB-月  |

|  | gp3                            | gp2 |
|--|--------------------------------|-----|
|  | 125 MiBs，每月佈建 MiBs 超過 125 MiBs |     |

### ⚠ Important

即使 gp3 磁碟區有不同的容量和效能成本，如果 gp3 磁碟區設定為相同的效能等級，則其一律比 gp2 磁碟區便宜。

下表顯示可透過將 gp2 轉換為 gp3 磁碟區以各種容量和效能組態達成的成本節省範例。

#### gp2 組態的範例

| 磁碟區大小 (GiB) | 最大 IOPS | 輸送量 (MiBs) | 成本 (美元/月) |
|-------------|---------|------------|-----------|
| 30          | 3000    | 128        | 3.00 美元   |
| 100         | 3000    | 128        | 10.00 美元  |
| 500         | 3000    | 250        | 50.00 美元  |
| 1000        | 3000    | 250        | 100.00 美元 |
| 2000        | 6000    | 250        | 200.00 美元 |
| 6000        | 16000   | 250        | 600.00 美元 |

#### gp3 (基準) 組態的範例

| 最大 IOPS | 輸送量 (MiBs) | 成本 (美元/月) | 成本降低 (相較於 gp2) |
|---------|------------|-----------|----------------|
| 3000    | 125        | 2.40 美元   | 20%            |
| 3000    | 125        | 8.00 美元   | 20%            |

| 最大 IOPS | 輸送量 (MiBs) | 成本 ( 美元/月 ) | 成本降低 ( 相較於 gp2) |
|---------|------------|-------------|-----------------|
| 3000    | 125        | 40.00 美元    | 20%             |
| 3000    | 125        | 80.00 美元    | 20%             |
| 3000    | 125        | 160.00 美元   | 20%             |
| 3000    | 125        | 480.00 美元   | 20%             |

### gp3 (gp2 比對 ) 組態的範例

| 最大 IOPS | 輸送量 (MiBs) | 成本 ( 美元/月 ) | 成本降低 ( 相較於 gp2) |
|---------|------------|-------------|-----------------|
| 3000    | 128        | 2.52 美元     | 16%             |
| 3000    | 128        | 8.12 美元     | 19%             |
| 3000    | 250        | 45.00 美元    | 10%             |
| 3000    | 250        | 85.00 美元    | 15%             |
| 6000    | 250        | 180.00 美元   | 10%             |
| 16000   | 250        | 550.00 美元   | 8%              |

如需成本分析，請參閱 Amazon EBS 資源中的 EBS gp2 到 gp3 遷移成本節省計算器一節。<https://aws.amazon.com/ebs/resources/>您可以下載計算器，並使用它來了解將 gp2 磁碟區遷移至 gp3 可以節省多少成本。

### 成本最佳化建議

如需如何完成遷移程序的指示，請參閱 AWS 儲存部落格上的將您的 [Amazon EBS 磁碟區從 gp2 遷移至 gp3](#)，並節省高達 20% 的成本文章。

## 其他資源

- [將您的 Amazon EBS 磁碟區從 gp2 遷移至 gp3，並節省高達 20% 的成本](#) (AWS 儲存部落格)
- [建置 AWS Config 自訂規則以最佳化 Amazon EBS 磁碟區類型](#) (AWS 雲端操作與遷移部落格)
- [透過刪除未使用的 Amazon EBS 磁碟區來控制 AWS 成本](#) (AWS 雲端操作與遷移部落格)
- [Amazon EBS 遷移公用程式](#) (GitHub)
- [尋找 2020 re : Invent 公告的節省成本](#) (AWS 雲端財務管理)
- [成本最佳化研討會](#) (AWS Well-Architected 實驗室)
- [gp2 到 gp3 遷移成本節省計算器](#) (下載)

## 修改 Amazon EBS 快照

### 概觀

刪除 EBS 磁碟區和管理快照的保留和封存，是從一開始控制成本的重要層面。您可以透過擷取 point-in-time 快照，將 EBS 磁碟區上的資料備份至 Amazon Simple Storage Service (Amazon S3)。快照是增量備份，因此只會在最近快照之後變更的裝置上儲存區塊。如此無須複製所有資料，可大幅減少建立快照所需的時間，並節省儲存成本。每個快照都包含將資料還原至新 EBS 磁碟區所需的所有資訊（從建立快照時開始）。

EBS 快照的費用是以 GB 月計算。您需要支付快照的大小，以及保留快照的時間長度的費用。定價因儲存層而異。對於 [Standard 方案](#)，您只需針對儲存的變更區塊付費。對於 Archive 方案，您需要支付所有存放快照區塊的費用。您還需要支付從 [Archive 層](#) 擷取快照的費用。以下是每個儲存層的範例：

- 標準方案 – 您有一個儲存 100 GB 資料的磁碟區。您需要為第一個快照（快照 A）的完整 100 GB 資料支付費用。在下一個快照（快照 B）時，您有 105 GB 的資料。然後，您只需為增量貼齊 B 額外支付 5 GB 的儲存空間。
- 封存層 – 您封存快照 B。然後快照會移至封存層，並向您收取完整的 105 GB 快照區塊費用。

您可以使用 [Amazon Data Lifecycle Manager](#) 來協助您設定生命週期，以按排程保留和管理快照。

### 成本影響

EBS 磁碟區和快照的費用會分開管理。EBS 快照的計費費率低於作用中的 EBS 磁碟區。當執行個體終止時，每個連接 EBS 磁碟區的 [DeleteOnTermination 屬性](#) 值會決定要保留或刪除磁碟區。根據預

設，根磁碟區的 `DeleteOnTermination` 屬性會設為 `True`。針對所有其他磁碟區類型，它會設為 `False`。這會建立運算子想要刪除 EC2 執行個體，但留下除了根磁碟區之外新增至執行個體的落後磁碟區的情況。如需有關檢查不再需要的磁碟區（及其相關聯的快照）的說明，請參閱 [Amazon EBS 文件中的檢視 Amazon EBS 磁碟區的相關資訊](#)。

根據預設，當您建立快照時，快照會存放在 Amazon EBS 快照標準層（標準層）中。存放在標準層中的快照是增量快照。這表示僅儲存最近快照後，磁碟區上已變更的區塊。[Amazon EBS 快照封存](#) 是新的儲存層，可用於低成本、長期儲存您很少存取的快照，不需要頻繁或快速擷取。從標準到存檔的定價差異很重要，應該是設定快照策略時的關鍵考量。對於計劃存放 90 天或更長時間且您很少需要存取的快照，Amazon EBS 快照封存可降低高達 75% 的快照儲存成本。

| Amazon EBS 快照儲存 | 成本            |
|-----------------|---------------|
| 標準              | 每月 0.05 USD   |
| 存檔              | 每月 0.0125 USD |

在較小的環境中，節省成本可能並不重要。大規模節省成本更為顯著，其中有多個帳戶和數千個 EC2 執行個體，即使 EBS 磁碟區已刪除，EBS 快照也會存在 TBs。

下表比較每月僅 50 TB 用量的標準層和封存層。即使規模較低，每年仍省下數千美元。

| Amazon EBS 快照儲存 | 每月成本      | 每年成本        |
|-----------------|-----------|-------------|
| 標準 50 TB        | 312.50 美元 | 3,750 美元    |
| 封存 50 TB        | 78.13 美元  | 937.60 美元   |
|                 | 年度節省      | 2,812.40 美元 |

## 成本最佳化建議

刪除快照可能不會降低您組織的資料儲存成本。其他快照可能會參考該快照的資料，被參考資料一律予以保留。例如，當您建立包含 10 GiB 資料之磁碟區的第一個快照，其快照大小也是 10 GiB。由於快照是增量式的，建立相同磁碟區的第二個快照時，僅會包含自從建立第一個快照後所有變動的資料區塊。第二個快照也會參考第一個快照中的資料。如果您變更 4 GiB 的資料並拍攝第二個快照，第二個快照的大小為 4 GiB。此外，第二個快照會參考第一個快照中未變更的 6 GiB。如需詳細資訊，請參閱

AWS 知識中心中的 [為什麼在刪除 EBS 磁碟區的快照後，再刪除磁碟區本身之後，我的儲存成本並未降低？](#)。

考慮下列各項：

- 您不需要支付另一個 AWS 帳戶擁有並與您的帳戶共用的快照費用。只有在您將共用快照複製到帳戶時，才會向您收取費用。您還需要為從共用快照建立的 EBS 磁碟區付費。
- 如果另一個快照（快照 B）參考快照（快照 A），則刪除快照 B 可能不會降低儲存成本。當您刪除快照時，只會移除該快照唯一的資料。其他快照所參考的資料仍會保留，而且您需為此參考的資料付費。若要刪除增量快照，請參閱 Amazon EBS 文件中的 [增量快照刪除](#)。

快照清理是執行工作負載時的標準操作實務 AWS。隨著時間的推移，快照最多可為您不需要的資料增加成本高昂的費用。

## 其他資源

- [透過刪除未使用的 Amazon EBS 磁碟區來控制 AWS 成本](#) (AWS 雲端操作與遷移部落格)
- [刪除 Amazon EBS 快照](#) (Amazon EBS 文件)
- [成本最佳化研討會](#) (AWS Well-Architected 實驗室)
- [使用 Amazon Data Lifecycle Manager 自動封存 Amazon EBS 快照](#) (AWS 儲存部落格)

## 刪除未連接的 Amazon EBS 磁碟區

### 概觀

未連接（孤立）EBS 磁碟區可能會導致 AWS 環境中不必要的儲存成本。請務必將未使用和未使用的 EBS 磁碟區定期檢閱和刪除納入 AWS 環境衛生的一部分。最佳實務是制定程序以持續檢閱 EBS 磁碟區的使用情況。您可以使用 [AWS Compute Optimizer](#) 來檢閱未充分利用的執行個體。本節可協助您識別、管理和刪除未連接或未充分利用的 EBS 磁碟區。

### Amazon EBS

[Amazon Elastic Block Store \(Amazon EBS\)](#) 是一種區塊層級裝置，可為 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體提供儲存磁碟區。EBS 提供持久性儲存體，可靈活地連接和分離 EC2 執行個體。這表示即使 EC2 執行個體終止，EBS 磁碟區的生命週期也會持續存在。[DeleteOnTermination](#) 屬性是一項功能，可控制在執行個體終止時保留或刪除連接的 EBS 磁碟

區。根據預設，根磁碟區的屬性會設為 True，導致刪除。False 其他磁碟區的設定為，因此會保留。

## 成本影響

未連接的 EBS 磁碟區也稱為未使用或孤立磁碟區，會根據佈建的儲存大小和儲存類型產生與連接磁碟區相同的費用。雖然 Amazon EBS 費用的平均成本似乎很低，每月每 GB 0.10 美元，但請務必了解未使用的 EBS 磁碟區累積可能會隨著時間產生大量成本。

例如，請考慮保留 50 個未使用 EBS 磁碟區的後果，每個磁碟區佈建的儲存大小為 100 GB，如下表所示。

| 儲存磁碟區數量 | 磁碟區類型          | 大小     | 每月總成本   |
|---------|----------------|--------|---|
| 50 個磁碟區 | gp2 (0.10 USD) | 100 GB | 100 GB 50.00 個 EBS 磁碟區月 0.10 USD = 500.00 USD |

上表中的案例每月可降低成本約 500 美元或每年 6,000 美元。這是降低成本的有效步驟。請務必將刪除未連接的 EBS 磁碟區納入 AWS 環境衛生的常規實務。

## 成本最佳化建議

您可以使用 AWS 輕鬆自動刪除未連接的 EBS 磁碟區。例如，您可以使用 AWS Lambda AWS Config、Amazon CloudWatch 和 AWS Systems Manager，根據存留期、標籤和其他規格定義刪除未連接磁碟區的條件。您也可以使用這些 AWS 服務功能大規模自動化清除程序。

為了避免意外後果，建議您在刪除未連接的 EBS 磁碟區之前執行盡職調查。

### 管理未連接的 EBS 磁碟區

建議您考慮遵循最佳實務：

- 符合合規要求 – 確認刪除未連接的 EBS 磁碟區符合組織的控管和合規要求。
- 設定資料備份和保留政策 – 刪除未連接的 EBS 磁碟區之前，請將任何重要資料備份到另一個儲存庫（例如 [Amazon S3](#)）。對於資料保留，[Amazon EBS 快照](#)是保留資料比 EBS 磁碟區更符合成本效益的方式，而且未來可以視需要還原磁碟區。如需有效管理快照的詳細資訊，請參閱本指南的[修改 Amazon EBS 快照](#)一節。

- 檢查相依性 – 檢查未連接 EBS 磁碟區和其他 AWS 資源之間的任何相依性。您可以使用 [AWS Management Console](#) 或 [API](#) 來收集 EBS 磁碟區的描述性資訊，例如大小、狀態和相關資源。這是防止刪除任何暫時未連接資源的重要步驟。
- 建立保留政策 – 為未連接的 EBS 磁碟區建立保留期間。這可協助您識別刪除未連接磁碟區的適當時間，確保您的 AWS 環境保持最佳化。例如，您可以建立 [Amazon EventBridge](#) 規則，以排程啟動 Lambda 函數。Lambda 函數可以使用 AWS SDK 主動識別任何未連接的 EBS 磁碟區、套用標記機制以輕鬆追蹤，以及在未連接的 EBS 磁碟區達到或超過定義的閾值時傳送通知。
- 標記未連接的 EBS 磁碟區 – [標記](#) EBS 磁碟區是一種有用的實務，可協助根據環境、應用程式或擁有者等屬性來組織和識別磁碟區。這在決定要刪除哪些未連接磁碟區時特別有用，因為它可讓您根據標籤快速識別不再需要的磁碟區。
- 確保安全刪除 – 檢閱 EBS 磁碟區上次連接的時間可協助您判斷刪除磁碟區是否安全。如需詳細資訊，請參閱 AWS 知識中心中的 [如何使用 AWS CLI 命令列出特定 Amazon EBS 磁碟區的附件或分離歷史記錄？](#)。
- 識別未充分利用的 EBS 磁碟區 – 識別和移除未充分利用的 EBS 磁碟區是高度建議的做法，可降低儲存成本和維護最佳化 AWS 的環境。AWS Trusted Advisor [AWS Compute Optimizer](#) 可協助您識別未充分利用的 EBS 磁碟區，並提供建議以降低成本並提高效率。例如，請參閱 [使用 \(GitHub\) 設定最佳化 EBS 磁碟區的自動化 AWS Trusted Advisor](#)、[建立 Trusted Advisor 組織 \(TAO\) 儀表板 \(AWS Workshop Studio\)](#)，以及 [使用 \(儲存體部落格\) 進行成本最佳化 Amazon EBS 磁碟區 AWS Compute Optimizer](#)。GitHub AWS

## 自動化清除未連接的 EBS 磁碟區

建議您考慮下列工具，以協助您自動清理未連接的 EBS 磁碟區：

- [AWS APIs\(DescribeVolumes\)](#) – 您可以使用 AWS SDKs 或 AWS Command Line Interface () 來篩選和尋找未連接的 EBS 磁碟區 AWS CLI。您可以使用指令碼或依排程執行的 [Lambda 函數](#) 來自動化此程序，以節省時間和精力。來自 GitHub 的 [範例指令碼](#) 示範了其運作方式。指令碼使用 Lambda 來分析 AWS CloudTrail 日誌並識別未連接的 EBS 磁碟區。
- [AWS Systems Manager 自動化](#) – 這可讓您自動化基礎設施中的例行維護和修復任務。若要開始使用，請[建立自動化 Runbook](#)，它定義一系列要以特定順序執行的步驟。例如，您可以建立 Runbook，先建立未連接 EBS 磁碟區的快照，然後刪除磁碟區本身。這可協助您自動化手動執行時耗時且容易出錯的任務。
- [AWS Config](#) – 這可讓您評估、稽核和追蹤 AWS 資源隨時間的變化。透過擷取組態變更，您可以使用 AWS Config 來評估環境中的合規、控管和資源使用率。例如，AWS Config 可以識別 [未使用的 EBS 磁碟區](#)。此外，您可以將 AWS Systems Manager 自動化與 [建立關聯 AWS Config](#)，以自動修復刪除未使用的 EBS 磁碟區。

## 其他資源

- [使用 AWS Config 和 \( 方案指引 \) 刪除未使用的 Amazon Elastic Block Store \(Amazon EBS\) 磁碟區](#) [AWS Systems Manager](#)
- [透過刪除未使用的 Amazon EBS 磁碟區來控制 AWS 成本](#) (AWS 雲端操作與遷移部落格)
- [AWSConfigRemediation-DeleteUnusedEBSVolume](#) (AWS Systems Manager 自動化執行手冊參考)

## Amazon FSx

Amazon FSx for Windows File Server 是針對 Windows 工作負載最佳化的全受管檔案儲存服務。它為您提供簡單且可擴展的解決方案，以執行以 Windows 為基礎的應用程式和工作負載，而無需複雜的儲存基礎設施管理。您可以使用 FSx for Windows File Server 輕鬆佈建和存取原生支援 Windows 應用程式的共用檔案儲存，包括 Microsoft SQL Server、Microsoft SharePoint 和自訂 .NET 應用程式。此外，FSx for Windows File Server 提供靈活的定價選項，例如 pay-as-you-go 和儲存配額，以及自動重複資料刪除，以減少儲存體使用量並最佳化效能和成本，協助您管理成本。

本節涵蓋下列主題：

- [選擇正確的 SMB 檔案儲存體](#)
- [在 Amazon FSx 中啟用重複資料刪除](#)
- [了解 FSx for Windows File Server 中的資料碎片](#)
- [了解 Amazon FSx 中的 HDD 磁碟區用量](#)
- [使用單一可用區域](#)

## 選擇正確的 SMB 檔案儲存體

### 概觀

AWS 提供各種全受管儲存服務，為您提供業界領先的檔案服務，同時結合最新的 AWS 基礎設施創新和安全性。您可以將服務整合 AWS 到基礎設施即程式碼 (IaC) 工作流程中，並將其與 AWS 運算、監控和資料保護服務整合。對於 Windows 工作負載，您可以從兩個全受管檔案服務中進行選擇，這些服務可用於滿足您的應用程式需求：FSx for Windows File Server 和 Amazon FSx for NetApp ONTAP。

## FSx for Windows File Server

Amazon FSx for Windows File Server 提供建置在 Windows Server 上的全受管共用儲存，並提供廣泛的資料存取、資料管理和功能。FSx for Windows File Server 可輕鬆與 Windows 環境整合，因為它是 Windows 原生服務。我們建議將 FSx for Windows File Server 用於使用者和群組共用、適用於 SQL Server 的 Always On 容錯移轉叢集執行個體、Windows 應用程式和虛擬桌面基礎設施 (VDI)。FSx for Windows File Server 也與 Amazon FSx File Gateway、Amazon Kendra、Amazon S3 的稽核日誌和 Amazon Data Firehose 完美整合。

## FSx for OnTAP

FSx for ONTAP 是以 NetApp 的專屬 ONTAP 檔案系統為基礎。它需要一定程度的提升技能，並且建議大多數現有的內部部署 NetApp 使用者使用。典型的使用案例包括使用者和群組共用、適用於 SQL Server 的 Always On 容錯移轉叢集執行個體，以及 Windows 應用程式。FSx for ONTAP 支援多個通訊協定，大於 64 TB 檔案系統（不含 DFS 命名空間伺服器的 PB 擴展）、複製、複寫、快照、壓縮（儲存效率）和智慧型資料分層。

## 成本影響

### FSx for Windows File Server

FSx for Windows File Server 是上 AWS 部署 SQL Server 容錯移轉叢集執行個體的第一個共用儲存解決方案。透過 FSx for Windows File Server，您可以使用 SQL Standard 版授權來啟動容錯移轉叢集執行個體。不過，這可以防止您依賴需要 SQL Server Enterprise Edition 授權的 Always On 可用性群組。透過從 SQL Server Enterprise Standard Edition 切換到 SQL Server Standard Edition，您可以節省 65-75% 的 [SQL Server 授權](#)。

您可以使用 FSx for Windows File Server for Failover Cluster Instances，從一般 EBS 儲存體卸載儲存 I/O。透過將 I/O 卸載至 FSx for Windows File Server，您可以縮減依賴高 Amazon EBS 輸送量和 IOPS 的 EC2 執行個體，而不會影響儲存輸送量。

### FSx for OnTAP

您可以使用 FSx for ONTAP 在區塊通訊協定 iSCSI 上執行 Microsoft 容錯移轉叢集，並從 SQL Server 即時檔案初始化、使用 SnapMirror 的跨區域複寫、防毒支援和複製中受益。如果您建立多個資料庫複本進行測試，複製可以大幅改善空間耗用以及建立這些資料庫複本的速度。此外，您可以使用 FSx for ONTAP，使用 NetApp SnapCenter 來管理 SQL Server 的 EC2 執行個體的備份、還原和複製功能。FSx for ONTAP 也提供從 SSD 到低成本容量集區儲存的自動分層，以混合效能和成本效益。

FSx for ONTAP 支援 NetApp 的檔案系統 (ONTAP)，與支援 Windows 原生 NTFS 檔案系統的 FSx for Windows File Server 不同。FSx for ONTAP 的最小大小為 1024 GB，而 FSx for Windows File Server 的起始大小可低至 32 GB。

### 與 Microsoft 分散式檔案系統的整合

FSx for Windows File Server 和 FSx for ONTAP 與 Microsoft [的分散式檔案系統 \(DFS\)](#) 整合，可無縫整合到現有的部署。規劃架構時，請記住下列事項：

- FSx for Windows File Server 和 FSx for ONTAP 在這兩種部署類型（多個可用區域和單一可用區域）上支援 [DFS 命名空間 \(DFS\)](#)。
- 只有適用於 Windows File Server 的 FSx 支援 [DFS 複寫 \(DFSR\)](#)，且僅在使用單一可用區域時。

### 成本最佳化建議

FSx for Windows File Server 和 FSx for ONTAP 的效能與定價非常相依。FSx for Windows File Server 定價主要取決於儲存容量和儲存類型、輸送量容量、備份和傳輸的資料。使用 FSx for ONTAP，您需要支付 SSD 儲存、SSD IOPS、容量集區使用量、輸送量容量和備份的費用。

| 檔案服務                        | 5 TB 儲存體的成本 | 組態   | 區域            |
|-----------------------------|-------------|--|---------------|
| FSx for Windows File Server | 982.78 美元   | 單一可用區域<br>SSD (15,000 IOPS)<br>32 MBps<br>5 TB 備份（不節省重複資料刪除成本）     | 美國東部 (維吉尼亞北部) |
| FSx for OnTAP               | 979.28 美元   | 單一可用區域<br>100% SSD<br>15,000 個讀寫容量層<br>15,000 SSD IOPS<br>128 MBps | 美國東部 (維吉尼亞北部) |

| 檔案服務 | 5 TB 儲存體的成本 | 組態                      | 區域 |
|------|-------------|-------------------------|----|
|      |             | 5 TB 備份 ( 不節省重複資料刪除成本 ) |    |

請謹記以下幾點：

- 重複資料刪除和壓縮可讓您縮減資料大小，將更多資料存放在實體裝置上，但您需要支付佈建的固態硬碟 (SSD) 或硬碟 (HDD) 儲存的費用。
- 您可以使用 FSx for ONTAP 來分層資料。您 100% 的資料很少會定期存取，且需要 SSD 儲存。您可以將冷且不常存取的資料移至容量層，以節省成本。
- 此處提及的價格是以 100% 的 SSD 層資料和 15,000 IOPS 的 SSD 層資料計算。

## 備份

根據預設，FSx for ONTAP 和 FSx for Windows File Server 都會將其全受管備份存放在 Amazon S3 上。不過，使用 FSx for ONTAP 時，還有使用 SnapVault 進行備份的額外選項，可將備份設定為位於容量層中。使用 SnapVault 備份是一種自我管理的機制，比預設的全受管備份選項更具成本效益。全受管備份選項為每月每 GB 0.05 USD。FSx for ONTAP (10 : 1 SSD 到容量集區儲存) 上的 SnapVault 備份為 \$0.03221 (0.9x0.0219+0.1x0.125)。

請謹記以下幾點：

- AWS 受管備份提供一小時的精細程度。[SnapVault](#) 可讓您最短五分鐘。
- 您可以使用 NetApp 的工具 ( 例如 CLI 和 API) 來設定 SnapVault 關係和快照複寫。
- 啟用 SnapVault 磁碟區上的 all 分層政策，以使用容量層做為備份資料的儲存體。
- SnapVault 目的地可以位於相同 AWS 區域、跨區域或內部部署。這通常是單一可用區域或多個可用區域檔案系統備份目的地。相較之下，AWS Backup 受到 Amazon S3 的區域彈性支援。

## 正確調整大小

您也可以透過正確調整大小和防止過度佈建，節省成本並充分利用您的檔案系統。

若要正確調整大小，請執行下列動作：

1. 根據資料識別您目前的需求。對於典型的 Windows 工作負載，您可以使用內建的作業系統工具，例如 [Performance Monitor](#)。

- 在效能監控中，使用下列計數器來衡量您目前的效能需求。擷取間隔設定為一秒，日誌大小上限為 1,000 MB 並啟用覆寫。

```
Logman.exe create counter PerfLog-Short -o "c:\perflogs\PerfLog-Long.blg" -f bincirc -v mmdhmm -max 1024 -c "\\LogicalDisk(*)\*" "\\Memory\*" "\.NET CLR Memory(*)\*" "\\Cache\*" "\\Network Interface(*)\*" "\\Paging File(*)\*" "\\PhysicalDisk(*)\*" "\\Processor(*)\*" "\\Processor Information(*)\*" "\\Process(*)\*" "\\Thread(*)\*" "\\Redirector\*" "\\Server\*" "\\System\*" "\\Server Work Queues(*)\*" "\\Terminal Services\*" -si 00:00:01
```

- 若要開始日誌擷取，請執行 `logman start PerfLog-Short` 命令。若要停止日誌擷取，請執行 `logman stop PerfLog-Short` 命令。

#### Note

您可以在執行擷取的伺服器上的 `c:\perflogs` 中找到效能日誌檔案。如需詳細資訊，請參閱 Microsoft 文件中的 [Windows 效能監視器概觀](#)。

- 識別正確的組態後，請使用 Microsoft [DISKSPD](#) 等磁碟壓力工具，在 Amazon FSx 檔案系統上測試您的預估值是否正確。
- 如果您對效能感到滿意，請切換到檔案共享。

我們建議採用保守的儲存容量方法，因為它只能向上擴展。輸送量容量可以視需要縱向擴展和縮減。

## 其他資源

- [Amazon FSx for NetApp ONTAP FAQs](#) AWS (網站)
- [使用新指標最佳化 Amazon FSx for Windows File Server 效能](#) (AWS 儲存部落格)

## 在 Amazon FSx 中啟用重複資料刪除

### 概觀

重複資料刪除是一項功能，可讓您更有效率地存放資料，並減少容量需求。它涉及在資料中尋找和移除重複項目，而不會影響其真實性或完整性。重複資料刪除使用子檔案變數大小區塊和壓縮，為一般檔案伺服器提供 2:1 的最佳化比率，為虛擬化資料提供高達 20:1 的最佳化比率。重複資料刪除比 NTFS 壓縮更有效。重複資料刪除架構中固有的彈性是硬體故障期間的彈性，可對資料和中繼資料進行完整檢查總和驗證，包括中繼資料備援和最常存取的資料區塊。

FSx for Windows File Server 完全支援重複資料刪除。使用它可以為一般用途檔案共享平均節省 50–60%。在共享中，使用者文件節省範圍從 30–50%，軟體開發資料集節省範圍從 70–80%。請務必了解，透過重複資料刪除可以實現的儲存節省取決於資料集的性質，包括檔案之間存在多少重複。如果存放的資料本質上是動態的，則重複資料刪除不是很好的選項。

## 成本影響

為了因應企業中的資料儲存成長，管理員會合併伺服器，並制定容量擴展和資料最佳化關鍵目標。重複資料刪除的預設設定可以立即節省成本，或者管理員可以微調設定以查看額外的收益。例如，您可以將重複資料刪除設定為僅在特定檔案類型上執行，也可以建立自訂任務排程。

在高階，重複資料刪除有三種類型的任務：最佳化、垃圾收集和清理。請注意，除非您在最佳化後執行垃圾收集任務，否則不會釋放空間。您可以排程任務，也可以手動執行任務。手動啟動任務時，也可以使用排程重複資料刪除任務時可用的所有設定（排程特定設定除外）。

即使從重複資料刪除中有效節省了 25%，FSx for Windows File Server 仍大幅節省成本。這些預計節省的成本是以中的 [預估](#) 值為基礎 AWS 定價計算工具。

## 成本最佳化建議

FSx for Windows File Server 檔案系統上的重複資料刪除預設為未啟用。若要在 [PowerShell 上使用遠端管理](#) 來啟用重複資料刪除，您必須執行 Enable-FSxDedup 命令，然後使用 Set-FSxDedupConfiguration 命令來設定組態。如需詳細資訊，請參閱 [FSx for Windows File Server 文件中的管理檔案系統](#)。FSx

若要啟用重複資料刪除，請執行下列命令：

```
PS C:\Users\Admin> Invoke-Command -ComputerName amznfsxxxxxxx.corp.example.com -  
ConfigurationName FSxRemoteAdmin -ScriptBlock {Enable-FsxDedup }
```

若要驗證重複資料刪除組態，請執行下列命令：

```
Invoke-Command -ComputerName amznfsxxxxxxx.corp.example.com -ConfigurationName  
FSxRemoteAdmin -ScriptBlock {  
Set-FSxDedupSchedule -Name "CustomOptimization" -Type Optimization -Days  
Mon,Tues,Wed,Sat -Start 09:00 -DurationHours 7  
}
```

透過執行 PowerShell Measure-DedupFileMetadata cmdlet，您可以判斷如果您刪除一組資料夾、單一資料夾或單一檔案，然後執行垃圾收集任務，可以回收多少磁碟區的潛在磁碟空間。具體而言，

DedupDistinctSize值會告訴您，如果您刪除這些檔案，可以取回多少空間。檔案通常具有跨其他資料夾共用的區塊，因此重複資料刪除引擎會計算哪些區塊是唯一的，並在垃圾收集任務之後刪除。

預設的[重複資料刪除任務排程](#)設計為適用於建議的工作負載，且盡可能不具有侵入性（不包括針對備份用量類型啟用的優先順序最佳化任務）。如果工作負載有大量的資源需求，建議您排定任務僅在閒置時間執行，或減少或增加允許重複資料刪除任務使用的系統資源量。

根據預設，重複資料刪除會使用 25% 的可用記憶體。不過，您可以使用 `來增加此值-memory switch`。對於最佳化任務，我們建議您將範圍設定為 15 到 50。對於排程任務，您可以使用更高的記憶體耗用量。例如，使用垃圾收集和清理任務（通常排程在非上班時間執行），您可以設定更高的記憶體耗用量（例如 50）。

如需有關重複資料刪除設定的其他資訊，請參閱 FSx for Windows File Server 文件中的[使用重複資料刪除降低儲存成本](#)。

## 其他資源

- [了解重複資料刪除](#) (Microsoft 文件)
- [使用重複資料刪除來降低儲存成本](#) (FSx for Windows File Server 文件)

## 了解 FSx for Windows File Server 中的資料碎片

### 概觀

FSx for Windows File Server 效能取決於組態。它主要基於儲存類型、儲存容量和輸送量組態。您選取的輸送量容量會決定檔案伺服器可用的效能資源，包括網路 I/O 限制、CPU 和記憶體，以及檔案伺服器施加的磁碟 I/O 限制。您選取的儲存容量和儲存類型會決定儲存磁碟區可用的效能資源 - 儲存磁碟所施加的磁碟 I/O 限制。除了效能之外，組態選擇也會影響成本。FSx for Windows File Server 定價主要取決於儲存容量和儲存類型、輸送量容量、備份和傳輸的資料。

如果您有相對較大的檔案儲存和效能需求，則可以受益於資料碎片。資料分片涉及[將您的檔案資料分割](#)為較小的資料集（碎片），並將其儲存在不同的檔案系統中。從多個執行個體存取資料的應用程式，可以平行讀取和寫入這些碎片，以達到高水準的效能。同時，您仍然可以在一般命名空間下向應用程式顯示統一檢視。此外，它也可以協助將檔案資料儲存擴展到超過每個檔案系統支援 (64 TB) 的大型檔案資料集，最多可達數百 PB。

### 成本影響

對於大型資料集，部署多個適用於 Windows File Server 的小型 FSx 檔案系統通常更有效，而不是部署一個大型 SSD 共用，以實現相同層級的效能。結合使用 FSx for Windows File Server HDD 和 SSD

儲存類型可節省更多成本，並可讓您將工作負載與最佳的基礎磁碟子系統配對。在下表中，您可以看到單一 17 TB 檔案系統之間的差異，並將其與多個較小的檔案系統進行比較，這些檔案系統會新增至相同的容量。

### 具有多個工作負載的大型 SSD 檔案系統

| 伺服器名稱                              | 成本        | 組態  | 區域            |
|------------------------------------|-----------|---|---------------|
| Amazon FSx for Windows File Server | 5,716 USD | 17 TB SSD<br>30% 重複資料刪除<br>256 Mbps<br>17 TB 備份 | 美國東部 (維吉尼亞北部) |

### 使用 DFSN 的分割工作負載

| 伺服器名稱                              | 成本        | 組態  | 區域            | Share (分享) |
|------------------------------------|-----------|---|---------------|------------|
| Amazon FSx for Windows File Server | 1,024 USD | 2 TB SSD<br>20% 重複資料刪除<br>128 Mbps<br>2 TB 備份<br>Multi-AZ | 美國東部 (維吉尼亞北部) | 共用 1       |
| Amazon FSx for Windows File Server | 2,132 USD | 5 TB SSD<br>30% 重複資料刪除<br>256 Mbps<br>5 TB 備份             | 美國東部 (維吉尼亞北部) | 共用 2       |

| 伺服器名稱                              | 成本        | 組態  | 區域            | Share (分享) |
|------------------------------------|-----------|---|---------------|------------|
|                                    |           | Multi-AZ  |               |            |
| Amazon FSx for Windows File Server | 1,036 USD | 10 TB HDD<br>40% 重複資料刪除<br>128 Mbps<br>10 TB 備份<br>Multi-AZ | 美國東部 (維吉尼亞北部) | 共用 3       |
| DFSN Windows EC2 執行個體              | 27 USD    | t3a.medium<br>2 vCPUs<br>4 GiB 記憶體                          | 美國東部 (維吉尼亞北部) | DFSN 執行個體  |

大型 SSD 檔案系統的年成本為 68,592 美元。分割工作負載的年成本為 50,640 美元。在此範例中，工作負載與適當的後端儲存體相符時，可節省 26%。如需定價估算的詳細資訊，請參閱[AWS 定價計算工具](#)預估值。

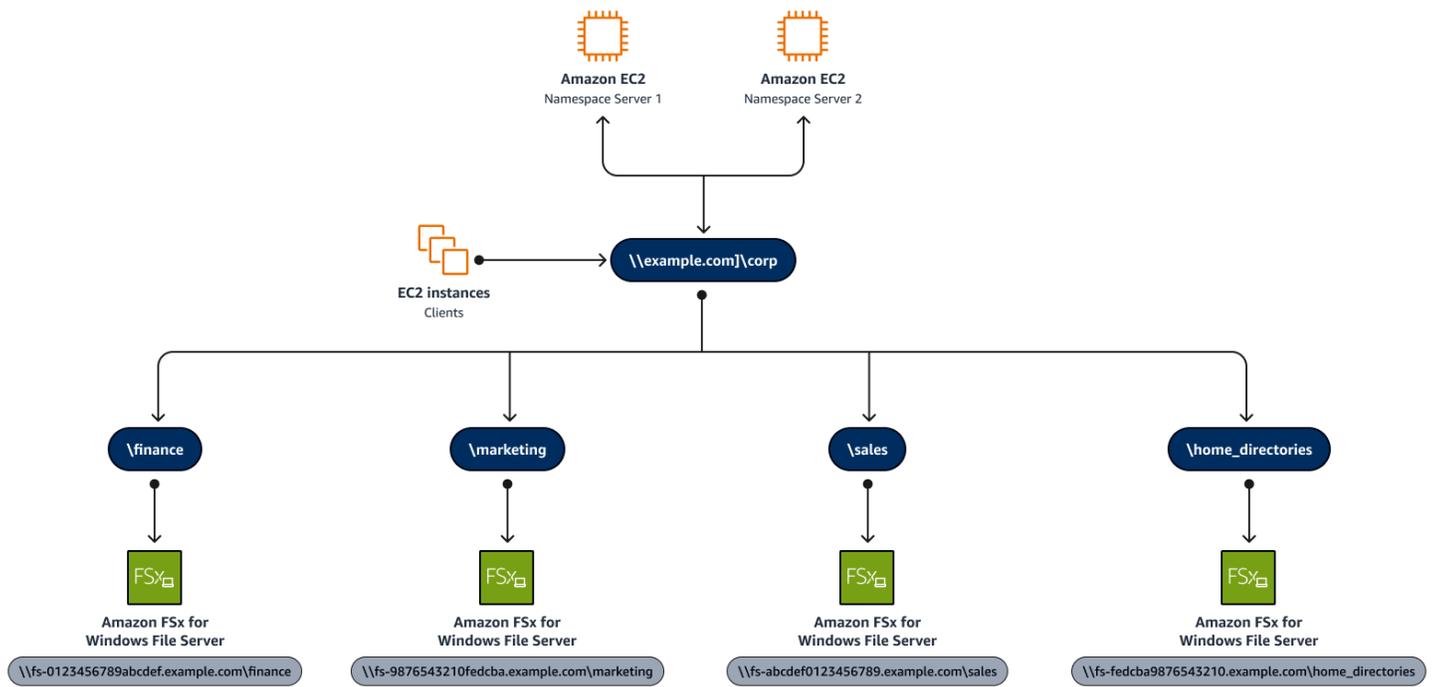
## 成本最佳化建議

若要部署重複資料刪除解決方案，您必須根據資料類型、I/O 大小和 I/O 存取模式來設定 [Microsoft DFS 命名空間](#)。每個命名空間總計支援最多 50,000 個檔案共享和數百 PB 的儲存容量。

它最有效率地選擇碎片慣例，將 I/O 平均分佈到您計劃使用的所有檔案系統中。監控您的工作負載將有助於進行其他最佳化或降低成本。如果您需要評估 Amazon FSx 檔案系統效能資訊的說明，請參閱 [FSx for Windows File Server 文件中的 FSx for Windows File Server 效能](#)。FSx

選擇碎片策略後，您可以使用 DFS 命名空間將檔案系統分組，以便輕鬆存取共享。這可讓使用者查看一個同質檔案系統，而實際上，他們使用專門建置的使用案例來存取各種不同的檔案系統。請務必使用適當的命名慣例建立共用，讓您的最終使用者可以輕鬆解密共用設計的目標工作負載。標記生產和非生產共用也很重要，因此最終使用者不會錯誤地將檔案放在錯誤的檔案系統中。

下圖顯示如何使用單一 DFS 命名空間做為多個 Amazon FSx 檔案系統的存取點。



請謹記以下幾點：

- 您可以將現有的 FSx for Windows File Server 共用新增至 DFS 樹狀目錄。
- Amazon FSx 無法新增至 DFS 共用路徑的根目錄。您只有一個子資料夾。
- 您必須部署 EC2 執行個體，才能提供 DFS 命名空間組態。

如需 DFS-N 組態的詳細資訊，請參閱 Microsoft 文件中的 [DFS 命名空間概觀](#)。如需使用 DFS 命名空間的詳細資訊，請參閱 YouTube 上的 [搭配使用 DFS 命名空間與 Amazon FSx for Windows File Server](#) 影片。

## 其他資源

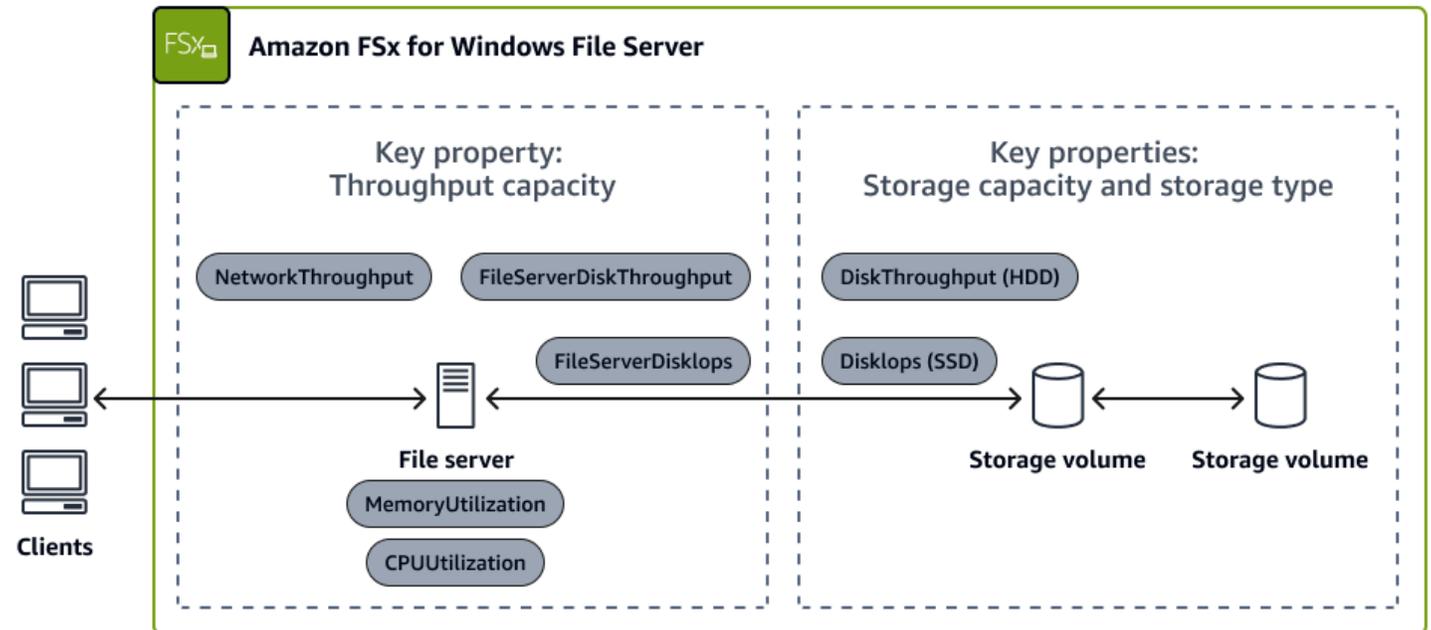
- [使用 DFS 命名空間分組多個檔案系統](#) (Amazon FSx 文件)
- [演練 6：使用碎片擴展效能](#) (Amazon FSx 文件)
- [搭配 Amazon FSx for Windows File Server \(實驗室\) 使用 DFS 命名空間](#) AWS

# 了解 Amazon FSx 中的 HDD 磁碟區用量

## 概觀

Amazon FSx for Windows File Server 提供獨立於檔案系統容量的彈性選擇輸送量。有兩種容量設定可用：HDD（硬碟）和 SSD（固態硬碟）。[EBS st1 磁碟機](#)用於 HDD 中的檔案系統儲存。[EBS io1 磁碟機](#)用於 SSD。

下圖顯示輸送量與儲存體設定之間的關係。



使用 HDD 型儲存時，您會收到 12 IOPS 基準，其中包含 80 個高載磁碟 IOPS（每 TiB 儲存的 IOPs），以及每秒 12 MB 的輸送量，以及每秒 80 個高載 MB（每 TiB 儲存）。例如，如果您的共享大小為 50 TB，則輸送量和 IOPS 的基準值為  $50 * 12 = 600$ 。

Amazon FSx for Windows File Server 提供 80 個高載 IOPS。當您的使用率低於基準速率時，系統會自動重新填充爆量額度，並在使用率高於基準速率時自動使用。例如，如果您的工作負載只使用 10 IOPS/TB 一小時（比基準速率低 2 IOPS/TB），則您可以在下一個小時使用 14 IOPS/TB（比基準高 2 IOPS/TB），然後再次用完爆量額度。

對於檔案操作，Amazon FSx for Windows File Server 提供 SSD 儲存的一致次毫秒延遲，以及 HDD 儲存的單一位數毫秒延遲。對於所有檔案系統，包括具有 HDD 儲存體的檔案系統，Amazon FSx for Windows File Server 會在檔案伺服器上提供快速（記憶體內）快取，因此無論儲存體類型為何，您都可以獲得主動存取資料的高效能和低於一毫秒的延遲。

適當時，使用 HDD 儲存有助於降低整體儲存容量的成本，並提供可靠的儲存平台以滿足您的需求。

## 成本影響

Amazon FSx for Windows File Server 效能取決於三個因素：儲存容量、儲存類型和輸送量。網路 I/O 效能和記憶體內快取大小完全取決於輸送量容量，而磁碟 I/O 效能則取決於輸送量容量、儲存類型和儲存容量的組合。

雖然建議將 SSD 用於 I/O 密集型工作負載，但有各種工作負載的需求可透過 HDD 效能規格來滿足。HDD 儲存體專為廣泛的工作負載而設計，包括主目錄、使用者和部門共享，以及內容管理系統。例如，如果您的使用者只需要對支援目前專案的資料進行低延遲存取，則您儲存的大部分資料不常存取。

您可以使用 [AWS 定價計算工具](#) 來比較 20 TB SSD 與 中的 HDD 檔案系統 us-east-1。如下表所示，即使沒有重複資料刪除的節省，在比較 HDD 檔案系統與 SSD 檔案系統時，成本差異也很重要。

| Amazon FSx 檔案系統組態            | 每月成本        |
|------------------------------|-------------|
| 20 TB 多可用區域 SSD (us-east-1 ) | 4,699.30 美元 |
| 20 TB 多可用區域 HDD (us-east-1 ) | 542.88 美元   |
| 預估每月節省                       | 4,156.42 美元 |

### Note

如需節省 Windows File Server 的額外 FSx，請參閱本指南的在 [Amazon FSx 中啟用重複資料刪除](#) 一節。

透過正確識別效能需求，您可以為工作負載選擇正確的儲存體，並降低成本。

## 成本最佳化建議

如果您決定使用 HDD 儲存體，請測試您的檔案系統，以確保其符合您的效能需求。與 SSD 儲存體相比，HDD 儲存體的成本較低，但磁碟輸送量和每單位儲存體的磁碟 IOPS 較低。它可能適用於具有低 I/O 需求的一般用途使用者共用和主目錄、不常擷取資料的大型內容管理系統，或具有少量大型檔案的資料集。

現有檔案系統的儲存類型無法變更。若要轉換 Amazon FSx for Windows File Server 檔案系統的儲存類型，您必須備份現有的檔案系統，並將其還原至具有所需儲存類型的新檔案系統。如果您想要將現有的 SSD 檔案系統轉換為 HDD 檔案系統，請注意，HDD 的最低容量要高出 2 TB。

若要還原不同儲存類型的備份，請執行下列動作：

1. [備份現有的檔案系統](#)。
2. 使用 HDD 儲存類型[建立新的 Amazon FSx 檔案系統](#)。
3. 將備份還原至具有所需儲存類型的新檔案系統。
4. 確認新的檔案系統具有正確的儲存類型，且您的資料保持不變。

將變更移至生產環境之前，建議您分析 Amazon FSx 檔案系統的效能，並確認變更是否可接受。如需更多指引，請參閱 AWS Storage Blog 上的[使用新指標最佳化 Amazon FSx for Windows File Server 效能](#)文章。

## 其他資源

- [使用 Amazon FSx 最佳化成本](#) (Amazon FSx 文件)

## 使用單一可用區域

### 概觀

本節說明何時使用適用於 [Windows File Server 的 Amazon FSx](#) 的單一可用區域實作更有益。它涵蓋了移至單一可用區域的案例，可降低成本，同時仍可讓您使用 Amazon FSx for Windows File Server 做為受管檔案儲存服務。建議您為生產工作負載的 Amazon FSx 實作單一可用區域。這有助於確保您具有多個可用區域的備援。

### 成本影響

與多個可用區域實作相比，單一可用區域檔案系統提供大約 40% 的成本降低。使用多個可用區域檔案系統時，您每月需為 SSD 支付 0.230 USD，HDD 為每月 0.025 USD，而 SSD 為每月 0.130 USD，而單一可用區域檔案系統上的 HDD 為每月 0.013 USD。您可以使用 [來查看成本的比較](#)，並建立您自己的預估值 [AWS 定價計算工具](#)。

對於 10 TB 檔案系統，每月為多個可用區域支付約 1,200 USD，或為單一可用區域支付每月 680 USD 的差額。此 [範例](#) 使用 10 TB FSx for Windows File Server 檔案系統搭配 SSD。重複資料刪除的預估節省成本為 50%。整體而言，單一可用區域的進入成本較低，但隨附下一節涵蓋的一些注意事項。

## 成本最佳化建議

### 單一可用區域部署

為了確保單一可用區域適合，請考慮您自己的內部 SLAs，以便將資料存放在 FSx for Windows File Server。這需要了解您是否有 SLAs 可提供給客戶（內部和外部），以及 Amazon FSx 單一可用區域的三九個可用性是否仍可讓您符合這些 SLAs。具有單一可用區域的 FSx for Windows File Server 仍有 99.9% 的執行時間。多個可用區域的 Amazon FSx SLA 大於 99.99%。對於關鍵任務工作負載，我們建議您在單一可用區域使用多個可用區域，即使是額外的成本。

單一可用區域部署非常適合諸如 SQL Server 資料庫備份等工作負載。它們可以使用 HDD 層提供低成本儲存，同時仍為您提供一致的運作時間。如果您需要更高層級的生產工作負載可用性，例如高可用性的 SQL 伺服器或生產應用程式存取，則單一可用區域不適合您的工作負載。對於備份、非生產測試和開發環境，Amazon FSx 單一可用區域實作可以降低您的營運成本。

一種使用案例，其中 Amazon FSx 單一可用區域檔案系統運作良好，正處於使用多個 Amazon FSx 單一可用區域檔案系統的生產環境中，作為使用 Always On 可用性群組之高可用性 SQL Server 叢集中的每個伺服器儲存體。如需詳細資訊，請參閱 AWS Storage Blog [上的 上高可用性 SQL Server 部署的最佳化成本 AWS](#)。

### 多區域複製

如果您想要利用 Amazon FSx 的多區域複製，則使用單一可用區域檔案系統（僅單一可用區域檔案系統運作）降低成本的潛在選項是。您可以部署支援使用原生 Microsoft DFS-R 的 [單一可用區檔案系統](#)。DFS-R 能夠跨區域和多個網站自動複製資料。如需使用 Amazon FSx 設定 DFS-R 的詳細資訊，請參閱 Amazon FSx 文件中的 [使用 Microsoft 分散式檔案系統複製](#)。

另一個節省多區域成本的替代方案是使用 AWS Storage Gateway。這可讓您在另一個區域中實作 [Amazon FSx 檔案閘道](#)，以存取 Amazon FSx 的多區域。如需詳細資訊，請參閱本指南的 [AWS Storage Gateway](#) 一節。

如果您跨區域工作，您必須考慮跨區域資料流量的資料傳輸成本。跨區域移動的流量會產生 0.02 USD/Gb 的費用。因此，如果您的大量資料變更一致，這將增加您的整體成本。[例如](#)，1 TB 的資料傳輸約等於 20.48 美元。

### Maintenance window (維護時段)

如果您將單一可用區域與 Amazon FSx 搭配使用，則維護時段是關鍵考量。在維護時段期間，由於基礎 Windows Server 的例行軟體修補，Amazon FSx 檔案系統無法使用大約 20 分鐘。如果您使用檔案系統進行夜間備份，請相應地調整 Amazon FSx 維護時段，以避免在備份期間中斷。您可以在建立 Amazon FSx 檔案系統之後調整 [維護時段](#)。

## 其他資源

- [可用性和耐久性：單一可用區和多可用區檔案系統](#) (Amazon FSx 文件)
- [Amazon FSx for Windows File Server 定價](#) AWS (網站)

## AWS Storage Gateway

AWS Storage Gateway 是一種混合雲端儲存服務，可將內部部署環境與 AWS 雲端儲存連線。它可讓您無縫整合現有的現場部署基礎設施與 AWS，讓您能夠從雲端儲存和擷取資料，並在混合環境中執行應用程式。對於 Windows 工作負載，您可以使用 Storage Gateway，使用 SMB 和 NFS 等原生 Windows 通訊協定來存放和存取資料。您可以使用 Storage Gateway，AWS 透過使用內部部署硬體和軟體做為雲端的橋接，來降低在上執行 Windows 工作負載的相關成本。這可讓您利用的可擴展性和成本效益，AWS 而不必對現有的基礎設施進行重大變更。

在 Storage Gateway 的保護下，您會取得 Amazon S3 File Gateway、Amazon FSx File Gateway、磁帶閘道和磁碟區閘道。S3 檔案閘道和 FSx 檔案閘道最常用於 Microsoft 工作負載。

### Amazon S3 檔案閘道

[Amazon S3 File Gateway](#) 可讓您將檔案存放在 Amazon S3 中，同時透過使用傳統 SMB 共用提供存取給使用者。這提供了熟悉的使用者介面，並透過將資料存放在 Amazon S3 中並利用各種 Amazon S3 儲存層來協助降低成本。您可以使用 S3 Intelligent Tiering 實作 Storage Gateway，協助您自動將生命週期檔案移至最低成本的儲存層，以進一步降低成本。我們建議將 S3 File Gateway 用於向外擴展、唯讀存取、快速重複讀取（從快取）和資料庫傾印。通常不建議用於高效能或高可用性寫入、編輯檔案或部門共享。

### Amazon FSx 檔案閘道

使用 [Amazon FSx Windows 檔案系統時](#)，[Amazon FSx File Gateway](#) 也可以節省成本。FSx 您可以站立 FSx 檔案閘道，以提供對另一個區域中 Amazon FSx 檔案系統的本地化存取，以避免擁有兩個獨立檔案系統的成本。如果您有多個現場部署檔案伺服器，並想要合併這些伺服器以避免支付多個硬體裝置的費用，這也很有幫助。

## 成本影響

### Amazon S3 檔案閘道

設定 S3 檔案閘道非常簡單，因為您可以使用 Storage Gateway 的啟動精靈。您可以使用 AWS 環境中的 EC2 執行個體，在幾分鐘內部署閘道。設定閘道之後，您可以將 Storage Gateway 共用設定為可

透過 SMB 和 NFS 通訊協定存取。對於典型的 Windows 工作負載，您也可以使用此設定來利用 Active Directory 環境，並設定檔案共享的許可。您可以有效地將 Storage Gateway 整合到正常使用中，因為它將做為典型的 Windows 檔案共用。檔案和資料夾會儲存為物件，而 NTFS 存取控制清單 (ACLs) 會儲存為中繼資料。

下表將 10 TB 儲存體的成本與三個可用的儲存體選項進行比較：

- FSx for Windows File Server
- Amazon S3 檔案閘道
- Amazon Elastic Block Store (Amazon EBS)

如果您使用 Amazon S3，儲存 10 TB 儲存空間的價格會大幅降低，因為您可以將資料分割為各種用量方案。在定價預估中，S3 Intelligent Tiering 用於其定價彈性。這包括 S3 Standard 的 80%、不常存取的 10%，以及 Amazon S3 Glacier 的 10%。雖然您可以使用 S3 Glacier，但請務必設定適當的生命週期規則，以確保移至 S3 Glacier 的任何檔案都不需要立即存取。S3 Glacier 僅用於封存用途，而非一般存取用量。

| 儲存系統   | 儲存 10 TB 的成本                      | 區域            |
|--|-----------------------------------|---------------|
| FSx for Windows File Server<br>( 假設重複資料刪除節省 50%) | <a href="#">683.20 USD SSD</a>    | 美國東部 (維吉尼亞北部) |
| Amazon S3 檔案閘道                                   | <a href="#">449.51 USD 的智慧型分層</a> | 美國東部 (維吉尼亞北部) |
| Amazon EBS                                       | <a href="#">1, 335.69 USD GP3</a> | 美國東部 (維吉尼亞北部) |

考慮下列各項：

- 在 S3 Glacier 中，除非您使用 [RestoreObject](#) API 將物件還原回 Amazon S3，否則會收到一般 I/O 錯誤。我們建議您使用 Amazon CloudWatch Events 對此 I/O 錯誤使用通知。如此一來，您的營運團隊可以對使用者做出反應，在他們可能需要存取的檔案上收到此錯誤。如需這些錯誤的詳細資訊，請參閱 Amazon S3 File Gateway 文件中的 [錯誤：InaccessibleStorageClass](#)。
- 除了 S3 Glacier 的存取限制之外，Storage Gateway 上的 [每個物件/資料夾只允許 10 ACLs](#)。在您決定使用 Storage Gateway 之前，請確定您不需要超過 10 個 ACL 項目。

## Amazon FSx 檔案閘道

與 Amazon S3 File Gateway 類似，FSx File Gateway 可讓您存取長期保留資料的檔案系統。在 Amazon S3 檔案閘道中，資料位於 Amazon S3 中。對於 FSx File Gateway，您的資料位於 FSx for Windows File Server。雖然 FSx for Windows File Server 有多個可用區選項，但沒有多區域選項。如果您有全球公司或遠端辦公室，您可能需要提供地理位置上更接近最終使用者的共用儲存平台，以避免延遲。如果您要部署另一個 Amazon FSx 檔案系統，這會增加全新的 Amazon FSx for Windows File Server 檔案系統和必要儲存體的成本。若要避免建立全新的檔案系統和重複成本，您可以在次要區域中部署 FSx 檔案閘道。這可為使用者提供檔案的本地化存取，同時有助於降低整體成本。

| 儲存系統                               | 儲存 10 TB 的成本   | 區域            |
|------------------------------------|----------------|---------------|
| Amazon FSx for Windows File Server | 683.20 USD SSD | 美國東部 (維吉尼亞北部) |
| Amazon FSx 檔案閘道                    | 503.70 美元/單一閘道 | 美國東部 (維吉尼亞北部) |

### Note

上表中的價格是以 [Storage Gateway 定價](#) 為基礎。

請謹記以下幾點：

- FSx File Gateway 可協助您為多區域工作負載每月節省約 180 USD (或每年 2100 USD)。
- FSx File Gateway 的資料傳輸費用較低，因為它只需要快取定期存取的檔案，而不是完整的次要副本。
- 雖然您可以在不同區域中有兩個 FSx for Windows File Server 部署，並使用 AWS Backup 或 保持更新 AWS DataSync，但這兩個選項都不是近乎即時。

## 成本最佳化建議

### Amazon S3 檔案閘道

S3 File Gateway 提供儲存檔案的低成本選項，但實作和使用檔案系統的方式有一些問題需要考慮。例如，S3 File Gateway 需要使用虛擬機器來執行 Storage Gateway 軟體。在中 AWS，預設會使用

m5.xlarge 執行個體在 Amazon EC2 中部署 Storage Gateway。如果您想要降低內部部署儲存成本，您可以將 Storage Gateway 部署為虛擬化平台上的虛擬設備，例如 VMware 和 Hyper-V。

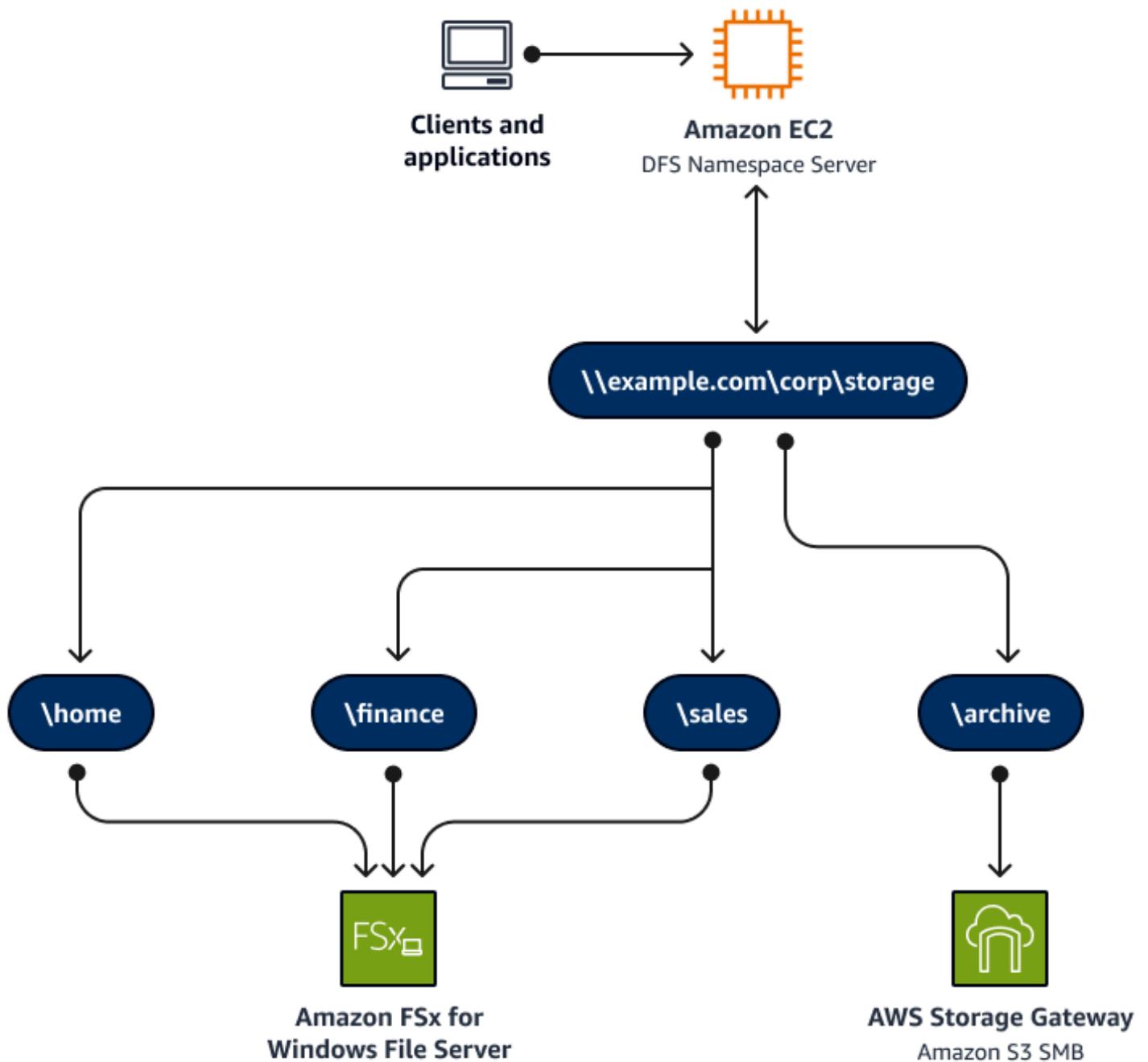
### 高可用性考量

執行 Storage Gateway 是存取檔案的單一失敗點。為了避免不必要的停機時間，我們建議您實作嚴格的存取控制，讓使用者可以變更或停止並啟動 Storage Gateway 執行個體。此外，對於上的部署 AWS，使用 Amazon Data Lifecycle Manager 建立路由快照以快速復原 Storage Gateway 實作會很有幫助。如果您使用 VMware 在內部部署執行 Storage Gateway，則可以將其設定為[高可用性](#)。

### 執行多個檔案系統

將每日使用的檔案工作負載與封存工作負載分開，可協助您避免不必要的儲存成本。Storage Gateway 能夠與 FSx for Windows File Server 檔案系統一起部署。透過使用[DFS 命名空間](#)，您可以呈現在 FSx for Windows File Server 上執行的主要每日使用儲存，以及在 Amazon S3 中執行的儲存（透過 Storage Gateway 存取）。

下圖顯示如何將單一 DFS 命名空間用作不同後端儲存選項的前端存取點。



用戶端會導向至資料夾結構，例如 `\\example.com\storage`。此主目錄包含子目錄。FSx for Windows File Server 檔案系統包含正常存取的檔案共用。您可以使用在 Storage Gateway 上建立的檔案共享來封存資料。使用者可以手動將項目封存至封存資料夾，或者您可以建置程序，將一些檔案從一般檔案共用自動移至封存資料夾。

考慮下列各項：

- 檢閱您的儲存需求，並提供足夠的[快取儲存空間](#)。
- 將閘道新增至 Active Directory 組態，並使用[標準 Windows ACLs 存取檔案](#)。

## FSx 檔案閘道

FSx File Gateway 的部署與 S3 File Gateway 的部署類似，但如果您使用啟動精靈，會更輕鬆。如需詳細說明，請參閱 [Amazon FSx 檔案閘道文件中的步驟 3：建立和啟用 Amazon FSx 檔案閘道](#)。在環境中部署 FSx File Gateway 之後，您可以將其與現有的 Amazon FSx 檔案系統建立關聯，並存取您的檔案。

部署 FSx 檔案閘道時，儲存是主要考量。預設儲存體提供 150 GB，這是快取檔案的適當空間。建立低可用空間的監控提醒有助於正確調整儲存大小，而不會過度配置。

## 其他資源

- [AWS Storage Gateway 資源](#) (AWS 文件)

# Active Directory

執行 Windows Server 的 Amazon Elastic Compute Cloud (Amazon EC2) 是部署 Windows 應用程式和工作負載的安全、可靠且高效能的環境。您可以快速佈建執行個體，並視需要擴展或縮減規模，同時只需支付使用的費用。Active Directory 服務是做為 Windows Server 環境中身分管理的主要來源。

本節涵蓋下列主題：

- [Amazon EC2 上的自我管理 Active Directory](#)
- [AWS Managed Microsoft AD](#)
- [AD Connector](#)

## Amazon EC2 上的自我管理 Active Directory

### 概觀

本節提供在 Amazon Elastic Compute Cloud (Amazon EC2) 上執行 Active Directory 的成本降低建議。主要重點是確定您可以適當地調整 Active Directory 網域控制站的大小，並使用的彈性 AWS 雲端來調整環境所需的調整。AWS 可協助您輕鬆停止執行個體，並調整大小以符合不斷變化的需求，或者如果您的擴展速度太快，則縮減執行個體的大小。選擇正確的執行個體大小和類型可以大幅節省成本。

### 成本影響

下表顯示透過一般用途執行個體選擇爆量執行個體系列執行個體之間的差異。此選項每月可以為您節省大量金錢。適當規劃和調整執行個體大小可協助您管理成本。

| 執行個體類型     | 執行個體的數目 | vCPU | 記憶體 | 成本           |
|------------|---------|------|-----|--------------|
| t3a.medium | 2       | 2    | 8   | 每月 81.76 美元  |
| m5a.large  | 2       | 2    | 8   | 每月 259.88 美元 |

如需成本的詳細資訊，請參閱 AWS 定價計算工具 [預估](#)。

每月節省 178.12 美元，最終您的網域控制站每年節省超過 2,000 美元。請記住，在一個帳戶中只有兩個網域控制站的小足跡。使用多個帳戶和其他網域控制站進行擴展時，此類節省可能會大幅降低成本。

## 成本最佳化建議

Microsoft 會在您部署 Active Directory 環境時提供[容量規劃建議](#)。我們建議您在規劃或擴展 Active Directory 環境時，將下列主要元件納入考量：

- 記憶體
- 網路
- 儲存
- 處理器

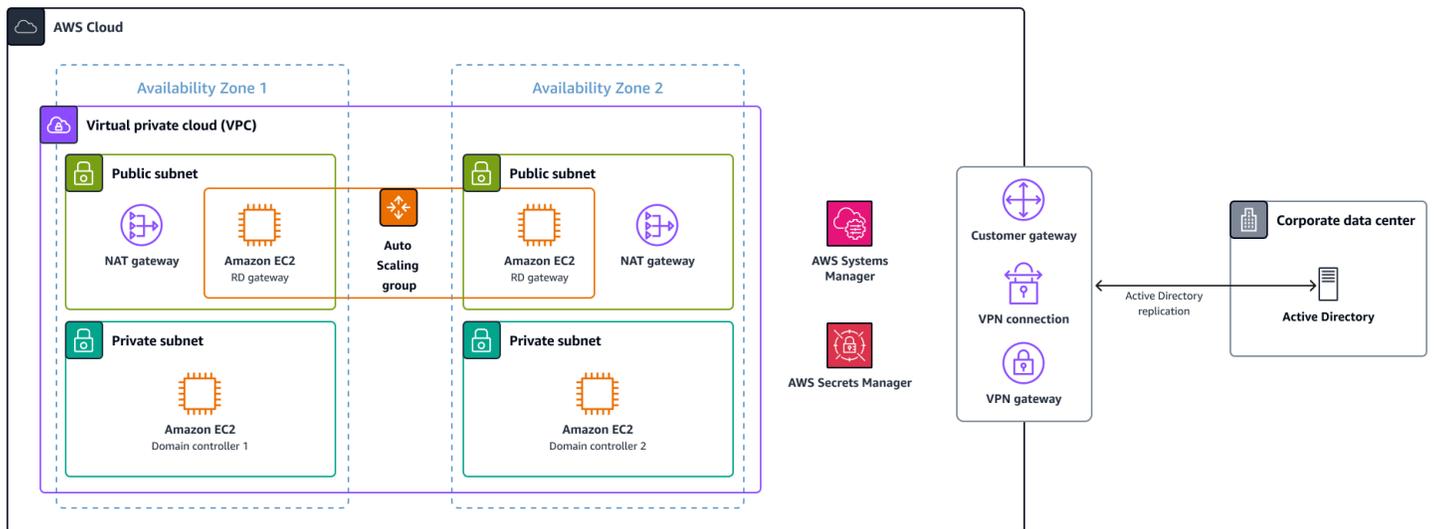
在記住這些主要元件時，您可以選取對 Active Directory 環境有意義的執行個體類型 AWS。本節涵蓋 AWS 幾個部署案例的範例 Active Directory。如果您不打算處理與現場部署環境相同的使用者和電腦數量 AWS，這些案例會明確指出您不需要在中複寫現場部署環境。

下表重點介紹有關 vCPU、記憶體和磁碟的重要元件，以滿足您的 AWS 足跡。

| 元件        | 估算                           |
|-----------|------------------------------|
| 儲存體/資料庫大小 | 每個使用者 40–60 KB               |
| RAM       | 資料庫大小<br>基礎作業系統建議<br>第三方應用程式 |
| 網路        | 1 GB                         |
| CPU       | 每個核心有 1,000 個並行使用者           |

### 混合部署案例

下圖顯示 Active Directory 混合部署的範例架構。



如圖表所示，您通常具有內部部署足跡，然後將其擴展到中 AWS 雲端。在遷移的初始階段，您通常不會在其中部署所有使用者和伺服器 AWS。這就是為什麼一開始部署較小大小的足跡很重要，以節省遷移工作的成本。

如果您要使用伺服器和使用者在內部部署驗證來維護內部部署足跡，則不需要網域控制站使用相同的足跡 AWS。透過遵循 Active Directory 最佳實務，您可以實作適當的 [Active Directory 網站和服務](#)，將使用者和電腦驗證為您的內部部署足跡，同時只驗證您的 AWS 足跡給其中的網域控制站 AWS。這可讓您避免在上過度調整 Active Directory 佔用空間，AWS 方法是將使用限制為僅 AWS 資源，而不是所有內部部署基礎設施。如需設計混合式設定的指引，請參閱 Microsoft 文件中的 [網域控制站和網站考量的適當放置](#)。

## 透過正確調整大小來最佳化 AWS 遷移

如果您要為使用者部署新的 Active Directory 執行個體，或計劃 AWS 針對 Active Directory 基礎設施完全遷移至 AWS，建議您針對上表中的執行個體選擇，根據 Microsoft 對 vCPU、記憶體和磁碟空間的建議來規劃大小。

如果這是新的足跡，您可以啟動小型的執行個體，並利用輕鬆 [變更執行個體類型](#) 的能力，在環境持續成長時調整其大小 AWS。本指南的 [Windows on Amazon EC2](#) 一節說明如何監控和檢閱 CPU 和記憶體使用率 AWS。如此一來，您就可以知道何時增加 EC2 執行個體的大小。

如果您要將內部部署 Active Directory 環境完全遷移至 AWS，您可以實作相同的大小調整計劃，以確保適當的效能。在複寫現場部署的內容之前 AWS，我們建議您完成 Active Directory 環境的徹底檢閱。這可協助您避免過度佈建。請務必使用效能監視器來收集現有網域控制站的流量和使用率的相關資訊。這可讓您了解整體用量，以便正確調整大小並最終降低成本。

## 在上最佳化 Active Directory AWS

如果您正在執行 Active Directory AWS，也請務必持續監控使用率，並視需要變更執行個體大小，以減少支出。您可以使用 AWS Compute Optimizer 來取得您正在執行之資源的相關資訊 AWS。如需有關使用 Compute Optimizer 正確調整 Windows 工作負載大小的資訊，請參閱本指南的 [Windows on Amazon EC2](#) 一節。如需更全面的深入探討，您可以使用效能監視器來監控 Active Directory 網域控制站的使用率、評估效能，然後相應地調整大小。

您也可以使用 CloudWatch 來監控網域控制站的效能。若要最佳化網域控制站（向上或向下擴展），您可以使用 CloudWatch 中可用的指標，協助您做出正確的決策。您可以使用 CloudWatch 代理程式來設定要傳送以進行資料收集的自訂效能監控指標。如需說明，請參閱 AWS 知識中心的 [如何使用 CloudWatch 代理程式在 Windows 伺服器上檢視效能監控的指標？](#)。

部署 CloudWatch 代理程式之後，您可以在 下的代理程式組態檔案中設定下列指標 `metrics_collected`：

| 指標類別                     | 指標名稱      |
|--------------------------|-----------|
| 資料庫到執行個體 (NTDSA)         | 資料庫快取命中 % |
| I/O 資料庫讀取平均延遲            |           |
| I/O 資料庫讀取/秒              |           |
| I/O 日誌寫入平均延遲             |           |
| DirectoryServices (NTDS) | LDAP 繫結時間 |
| DRA 待定複寫操作               |           |
| DRA 待定複寫同步               |           |
| DNS                      | 遞迴查詢/秒    |
| 遞迴查詢失敗/秒                 |           |
| 每秒收到的 TCP 查詢             |           |
| 收到的查詢總數/秒                |           |
| 傳送的總回應數/秒                |           |

| 指標類別             | 指標名稱          |
|------------------|---------------|
| 每秒收到的 UDP 查詢     |               |
| LogicalDisk      | 平均磁碟佇列長度      |
| % 可用空間           |               |
| 記憶體              | % 的遞交位元組使用中   |
| 長期平均待命快取生命週期 (s) |               |
| 網路介面             | 傳送的位元組數/秒     |
| 接收的位元組/秒         |               |
| 目前頻寬             |               |
| NTDS             | ATQ 估計佇列延遲    |
| ATQ 請求延遲         |               |
| DS 目錄讀取/秒        |               |
| DS 目錄搜尋/秒        |               |
| DS 目錄寫入/秒        |               |
| LDAP 用戶端工作階段     |               |
| LDAP 搜尋/秒        |               |
| LDAP 成功繫結/秒      |               |
| 處理器              | 處理器時間百分比      |
| 整個安全系統統計資料       | Kerberos 身分驗證 |
| NTLM 身分驗證        |               |

## 其他資源

- [Active Directory Domain Services on AWS：合作夥伴解決方案部署指南](#) (AWS 文件 )
- [Active Directory Domain Services 的容量規劃](#) (Microsoft 文件 )
- 在 [EC2 執行個體上執行 Active Directory 的設計考量](#) (AWS 白皮書 )

## AWS Managed Microsoft AD

### 概觀

AWS Directory Service for Microsoft Active Directory 也稱為 AWS Managed Microsoft AD，由 Windows Server Active Directory 提供支援，並由 管理 AWS。您可以使用 AWS Managed Microsoft AD 將範圍廣泛的 Active Directory 感知應用程式遷移到 AWS 雲端。AWS Managed Microsoft AD 可與各種原生 Active Directory 應用程式和服務搭配使用。它也支援 [AWS 受管應用程式和服務](#)。雖然由於 服務及其計費機制 AWS Managed Microsoft AD， 的成本最佳化槓桿不多，但有一些設計原則可協助您將成本降至最低。

### 成本影響

由於 AWS Managed Microsoft AD 是基於現有 SKUs 受管服務，因此調整大小是一個相對簡單的程序。目前有兩種大小調整 SKUs：標準版和企業版。其他 SKUs 包括目錄共用、新增其他網域控制站（包括其他區域），以及跨區域資料傳輸。

### 成本最佳化建議

AWS Managed Microsoft AD Standard Edition 和 AWS Managed Microsoft AD Enterprise Edition 之間存在差異。Enterprise Edition 最多支援 500,000 個 Active Directory 物件、125 個帳戶共用（軟性限制），並支援多區域。Standard Edition 最多支援 30,000 個 Active Directory 物件、5 個帳戶共用（軟性限制最多約 30 個），而且沒有多區域支援。

選擇目錄類型之前要考慮的問題如下：

- 是否需要多區域支援？
- 目錄是否會與超過 30 個帳戶共用？
- Active Directory 物件計數是否會超過 30,000 個？

如果上述任何問題的答案為是，則需要 Enterprise Edition。如果所有問題的答案為否，我們建議您從標準版開始。

#### Note

您可以將目錄從 Standard Edition 升級至 Enterprise Edition，但無法降級目錄。部署 Standard Edition 不會經過單向門。如果您想要將目錄升級至 Enterprise Edition，請聯絡 AWS。

當您在 Enterprise Edition 中 AWS Managed Microsoft AD 共用目錄時，每個共用都會產生費用。這低於在每個帳戶中部署目錄的成本，但請記住，如果不勾選，共享成本可能會上升。我們建議您只與包含 Amazon Relational Database Service (Amazon RDS) 和 Amazon FSx for Windows File Server 的帳戶共用目錄，因為只有這些服務支援此功能。請記住，您可以選擇將 FSx for Windows File Server 與自我管理的 Active Directory 整合，包括 AWS Managed Microsoft AD。如果另一個帳戶中只需要 Amazon FSx，則您可以針對 AWS Managed Microsoft AD 執行自我管理的 Amazon FSx 部署，而不需要共用目錄。

決定何時部署其他網域控制站時，請記住，僅 AWS Managed Microsoft AD 支援相同 VPC 中不同可用區域中的兩個子網路。新增其他網域控制站不允許您新增其他子網路。若要判斷您是否因為效能問題而必須新增其他網域控制站，請檢閱 [CloudWatch 中的網域控制站效能指標](#)。這會告訴您一個或所有網域控制站是否不堪重負。如果您判斷只有一個網域控制器不堪重負，新增其他網域控制器就無法減輕負載，而且您需要更深入探索應用程式，而不是在目前可用的網域控制器之間進行負載平衡。如果所有網域控制站都大量使用，新增額外的網域控制站可以減少現有網域控制站的負載。如需如何自動化擴展的說明，請參閱 AWS 安全部落格中的 [如何根據使用率指標自動化 AWS Managed Microsoft AD 擴展](#)。

如果您將目錄擴展到多個區域，建議您不要使用 NETLOGON 或 SYSVOL 共用目錄進行檔案儲存。所有網域控制站都會複寫這些共用的內容。不使用共用進行檔案儲存會將資料傳輸成本降至最低。

您也可以選擇使用註冊 Enterprise 協議 AWS。Enterprise Agreements 可讓您選擇量身打造最符合您需求的協議。如需詳細資訊，請參閱 [企業客戶](#)。

## 其他資源

- [AWS Managed Microsoft AD 配額](#) (AWS Directory Service 文件)
- [AWS Directory Service 定價](#) (AWS 網站)
- [上的 Active Directory 網域服務 AWS](#) (AWS 白皮書)

# AD Connector

## 概觀

[AD Connector](#) 是一項代理服務，可讓您輕鬆地將現有的現場部署 Microsoft Active Directory 連線至相容的[AWS 應用程式](#)，例如 Amazon WorkSpaces、Amazon QuickSight，以及 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體的無縫網域聯結，而無需在雲端快取任何資訊。您可以使用 AD Connector 將一個服務帳戶新增至 Active Directory。AD Connector 不需要進行目錄同步，也不需要託管聯合基礎設施的成本和複雜性。雖然由於服務的性質及其計費機制，AD Connector 的成本最佳化控制桿不多，但您可以遵循本節的設計建議，將成本降至最低。

## 成本影響

AD Connector 是一種以預設 SKUs 為基礎的受管服務。這使得調整大小的過程變得簡單。有兩種大小調整 SKUs：小型和大型。您可以使用 [AWS 定價計算工具](#) 進行涉及 AD Connector 的成本估算。

## 成本最佳化建議

除了後端運算資源之外，小型和大型連接器大小並無差異。

選擇目錄類型之前要考慮的問題如下：

- 是否有大量 (10,000+) 使用與 AD Connector 整合 AWS 的應用程式的作用中使用者？
- 使用者是否為許多、深度或循環巢狀群組的成員？

如果兩個問題的答案都是否，我們建議您從小號開始。如果您對上述任何問題回答是，則可能值得考慮大型問題。您可以從小尺寸 AD Connector 開始，如果目錄因效能而受損，您可以請求將目錄升級至大尺寸。

### Note

您可以將 AD Connector 從小型升級至大型，但 AD Connector 無法降級。

大多數效能問題與 AD Connector 無關，但內部部署 Active Directory 網域控制站因為許多使用者是許多、深度或循環巢狀群組的成員而不堪重負。

您也可以選擇使用註冊 Enterprise 協議 AWS。Enterprise Agreements 可讓您選擇量身打造最符合您需求的協議。如需詳細資訊，請參閱[企業客戶](#)。

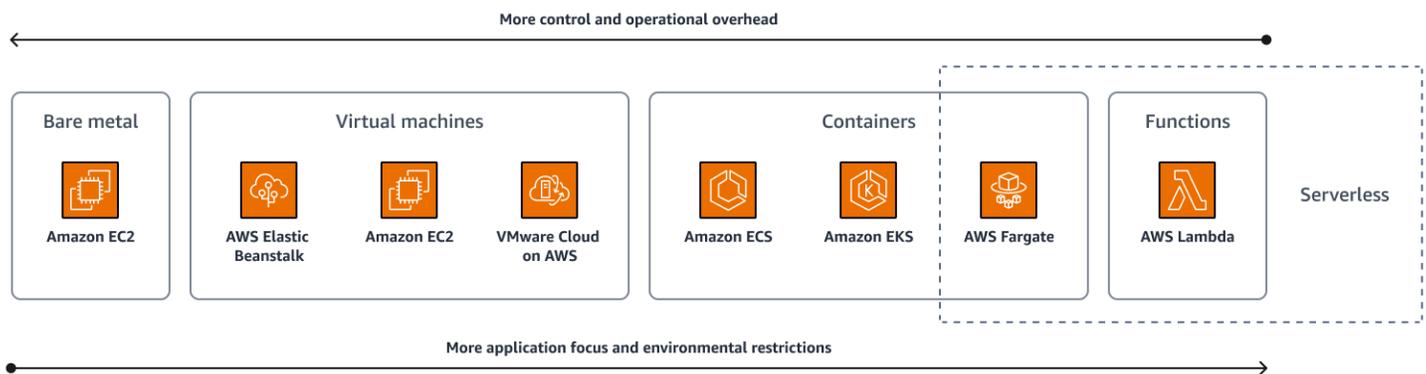
## 其他資源

- [AD Connector 配額](#) (AWS Directory Service 文件 )
- [其他目錄類型定價](#) (AWS 網站 )
- [上的 Active Directory 網域服務 AWS](#) (AWS 白皮書 )

# .NET

開發和部署 .NET 應用程式是協助您實現雲端運算提供之擴展和敏捷性的重要關鍵。對於許多舊版 .NET 應用程式，在中執行應用程式最合適的運算選擇 AWS 是使用虛擬機器，無論是透過 AWS Elastic Beanstalk 或 Amazon Elastic Compute Cloud (Amazon EC2)。您也可以 Windows 和 Linux 容器中執行 .NET 應用程式。

引進 .NET 核心可讓您設計現代化的 .NET 應用程式，以利用所有雲端優勢。現代應用程式可以使用傳統的運算選擇集，也可以鎖定各種類型的無伺服器環境，包括 AWS Fargate 或 AWS Lambda。 .NET 6+ 現在可在 Graviton2 EC2 系列等 ARM64 EC2 執行個體上提供工作負載的高效能託管。這可讓您存取 Amazon EC2 上最新一代的處理器。這表示您的應用程式可以託管在專門針對工作負載類型的運算上，例如視訊編碼、Web 伺服器和高效能運算 (HPC)。



本節提供建議，協助您調整 .NET 應用程式，以利用雲端的優勢，並專注於成本效益。

本節涵蓋下列主題：

- [重構現代 .NET 並移至 Linux](#)
- [容器化 .NET 應用程式](#)
- [使用 Graviton 執行個體和容器](#)
- [支援靜態 .NET Framework 應用程式的動態擴展](#)
- [使用快取來減少資料庫需求](#)
- [考慮無伺服器 .NET](#)
- [考慮專門建置的資料庫](#)

# 重構現代 .NET 並移至 Linux

## 概觀

現代化舊版 .NET Framework 應用程式可協助您改善安全性、效能和可擴展性。現代化 .NET Framework 應用程式的有效方法是將應用程式遷移至現代 .NET 版本 (6+)。以下是將這些應用程式移至開放原始碼 .NET 的一些主要優點：

- 在 Linux 作業系統上執行 Windows 授權成本以降低其成本
- 利用現代語言的可用性
- 取得最佳化以在 Linux 上執行的效能

許多組織仍在執行舊版的 .NET Framework。這可能會帶來安全風險，因為 Microsoft 不再處理舊版本的漏洞。Microsoft 已結束對 .NET Framework 4.5.2、4.6 和 4.6.1 的最新版本的支援。請務必評估繼續執行舊版架構的風險和好處。為了降低風險並降低成本，值得投入時間和精力來重構現代版本的 .NET。

## 成本影響

考慮一般用途的 EC2 執行個體類型 (m5)，可提供運算、記憶體和聯網資源的平衡。這些執行個體適用於各種應用程式，例如 Web 伺服器、中型資料庫和原始程式碼儲存庫。

例如，在美國東部（維吉尼亞北部）的 Windows Server（包含授權）上，具有 4 個 vCPUs 和 16 GB 記憶體的隨需 m5.xlarge 執行個體每月花費 274.48 USD。Linux 伺服器上的相同資源每月花費 140.16 美元。在此範例中，當您將應用程式從 .NET Framework 遷移到 .NET 的現代版本，並在 Linux 伺服器上執行應用程式時，成本會降低 49%。您的成本可能會因您在選取 [EC2 執行個體](#) 時選擇的選項（例如，執行個體類型、作業系統、儲存體）而有所不同。您可以使用 [Savings Plans](#) 或 [預留執行個體](#) 進一步最佳化成本。如需詳細資訊，請使用 [AWS 定價計算工具](#) 執行成本估算。對於包含 Windows 的執行個體，無論定價模式為何，授權費用為 [每小時每 vCPU 0.046 USD](#)。

將這些 .NET Framework 應用程式移植到現代 .NET 需要開發人員的努力。您必須評估您的應用程式及其相依性，以查看它們是否與目標平台版本相容。適用於 [AWS .NET 的 Porting Assistant](#) 是一項輔助工具，可掃描 .NET Framework 應用程式並產生 .NET 相容性評估，協助您更快地將應用程式移植到與 Linux 相容。適用於 .NET 的移植助理可識別與 .NET 的不相容、尋找已知的替代項目，並產生詳細的相容性評估。移植解決方案之後，您必須手動變更程式碼，才能成功編譯專案與相依性。這可減少將應用程式現代化為 Linux 所需的手動工作。如果您的應用程式支援 ARM 處理器，移至 Linux 會解鎖使用 Graviton 執行個體的能力。這可協助您在進一步降低成本時達到額外的 20%。如需詳細資訊，請參閱 AWS 運算部落格中的 [為 .NET 5 提供 AWS Graviton2：Benchmarks 的動力](#)。

還有其他工具，例如 [AWS Toolkit for .NET Refactoring](#) 和 [.NET Upgrade Assistant](#)，可協助您將舊版 .NET 架構應用程式移植到現代 .NET。

## 成本最佳化建議

若要遷移 .NET Framework 應用程式，請執行下列動作：

1. 先決條件 – 若要使用適用於 .NET 的移植助理，您必須在計劃分析應用程式原始碼的機器上安裝 .NET 5+。機器上的資源必須至少有 1.8 GHz 的處理速度、4 GB 的記憶體和 5 Gb 的儲存空間。如需詳細資訊，請參閱適用於 .NET 的移植助理文件中的[先決條件](#)。
2. 評估 – 將 .NET 的移植助理下載為[可執行檔](#)（下載）檔案。您可以在機器上下載並安裝工具，以開始評估您的應用程式。評估頁面包含與現代 .NET 不相容的移植專案、套件和 APIs。因此，您會在評估後在解決方案中收到建置錯誤。您可以將評估調查結果檢視或下載至 CSV 檔案。如需詳細資訊，請參閱[適用於 .NET 的移植助理文件中的移植解決方案](#)。
3. 重構 – 評估應用程式後，您可以將專案移植到目標架構版本。當您移植解決方案時，您的專案檔案和一些程式碼將由移植助理修改。您可以檢查日誌來檢閱原始程式碼的變更。在大多數情況下，程式碼需要額外的努力來完成遷移和測試，以使其生產準備就緒。視應用程式而定，某些變更可能包括實體架構、身分和身分驗證。如需詳細資訊，請參閱[適用於 .NET 的移植助理文件中的移植解決方案](#)。

這是將應用程式現代化為容器的第一步。可能有多個商業和技術驅動因素，將您的 .NET Framework 應用程式現代化為 Linux 容器。其中一個重要驅動因素是透過從 Windows 作業系統移至 Linux 來降低總擁有成本。如此可降低將應用程式遷移至跨平台版本的 .NET 和容器以最佳化資源使用率時的授權成本。

將應用程式移植到 Linux 之後，您可以使用 [AWS App2Container](#) 將應用程式容器化。App2Container 使用 Amazon ECS 或 Amazon EKS 作為您可以直接部署的端點服務。App2Container 提供所有必要的基礎設施即程式碼 (IaC) 部署成品，以重複容器化您的應用程式。

## 其他考量事項和資源

- 如果您的應用程式建置在 VB.NET (2002 年的傳統架構) 上，並想要將應用程式移植到 .NET 6，請參閱 AWS 部落格上的[使用適用於 .NET 的移植助理將舊版 VB.NET 應用程式移植到 .NET 6.0](#)。
- 如果您在 Windows Communication Foundation (WCF) 上有舊版應用程式，並想要在現代 .NET 上執行，則可以採用 CoreWCF。如需詳細資訊，請參閱 AWS 部落格上的 Microsoft 工作負載上的[使用適用於 .NET 的移植助理將舊版 WCF 應用程式現代化至 CoreWCF](#) 文章。

- 您可以將移植助理新增為 Visual Studio IDE 的延伸。這可讓您執行轉換程式碼所需的所有任務，而不需要在 IDE 和 .NET 的移植助理工具之間切換。如需詳細資訊，請參閱 AWS 部落格 Microsoft [工作負載上的使用 .NET Visual Studio IDE 擴充功能移轉助理加速 .NET 應用程式現代化](#) 文章。
- 適用於 [.NET 的移植助理現在是開放原始碼工具](#)，具有評估的原始程式碼和相容性分析元件。這可以鼓勵您的開發人員使用和共用 .NET 移植知識和最佳實務。
- 您可以使用 Toolkit for .NET Refactoring，將 AWS .NET 架構應用程式移植到 Linux 上的現代 .NET。如需詳細資訊，請參閱部落格 Microsoft [工作負載上的使用 AWS Toolkit for .NET Refactoring 文章加速 .NET 現代化](#)。AWS
- 您可以[加速容器化，並將 ASP.NET Core 應用程式遷移至 AWS 使用 AWS App2Container](#)。

## 容器化 .NET 應用程式

### 概觀

容器是一種輕量又有效率的方式，以一致且可重現的方式封裝和部署應用程式。本節說明如何使用 AWS Fargate 無伺服器容器服務，降低 .NET 應用程式的成本，同時提供可擴展且可靠的基礎設施。

### 成本影響

影響使用容器節省成本效率的一些因素包括應用程式的大小和複雜性、需要部署的應用程式數量，以及應用程式的流量和需求層級。對於小型或簡單應用程式，與傳統基礎設施方法相比，容器可能無法大幅節省成本，因為管理容器和相關服務的負荷實際上可能會增加成本。不過，對於較大或更複雜的應用程式，使用容器可以改善資源使用率並減少所需的執行個體數量，進而節省成本。

為節省成本，建議您在使用容器時謹記下列事項：

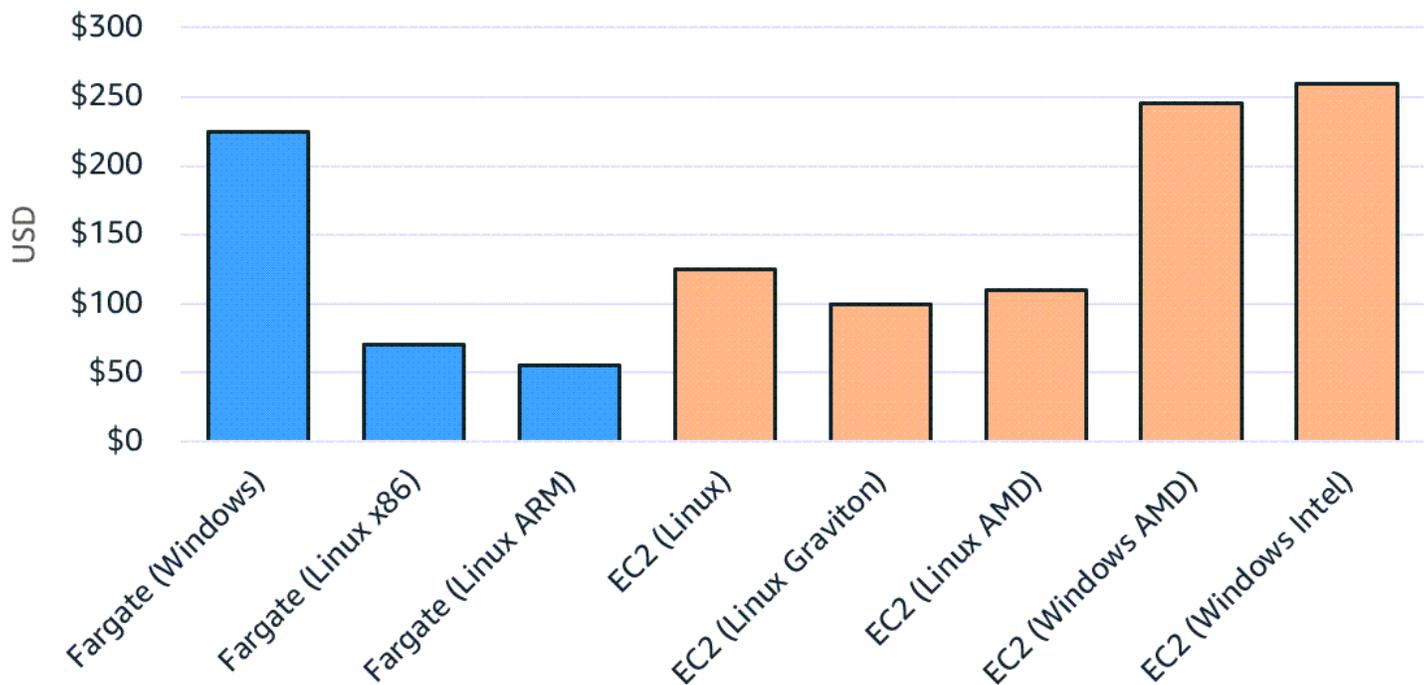
- 應用程式大小和複雜性 – 更大和更複雜的應用程式更適合容器化，因為它們往往需要更多資源，並且可以從改善的資源使用率中受益更多。
- 應用程式數量 – 您的組織必須部署的應用程式越多，容器化可節省的成本就越多。
- 流量和需求 – 遇到高流量和需求的應用程式可以從容器提供的可擴展性和彈性中受益。這可能會導致節省成本。

不同的架構和作業系統會影響容器成本。如果您使用的是 Windows 容器，由於授權考量，成本可能不會降低。Linux 容器的授權成本較低或不存在。下列圖表使用 AWS Fargate 美國東部（俄亥俄）區域中的基本組態，設定如下：每月 30 個任務，每個執行 12 小時，配置 4 vCPUs 和 8 GB 記憶體。

您可以從兩個主要運算平台中選擇在下列位置執行容器 AWS：以 [EC2-based 容器主機和無伺服器](#) 或 [AWS Fargate](#)。如果您使用 Amazon Elastic Container Service (Amazon ECS) 而非 Fargate，則必須維持執行中的運算（執行個體），以允許置放引擎在需要時執行個體化容器。如果您改用 Fargate，則只會佈建所需的運算容量。

下圖顯示使用 Fargate 與 Amazon EC2 的同等容器的差異。由於 Fargate 的彈性，應用程式的任務每天可以執行 12 小時，在非上班時間不會使用。不過，對於 Amazon ECS，您必須使用 EC2 執行個體的 [Auto Scaling 群組](#) 來控制運算容量。這可能會導致容量一天執行 24 小時，最終可能會增加成本。

## Monthly costs of Fargate and Amazon EC2



## 成本最佳化建議

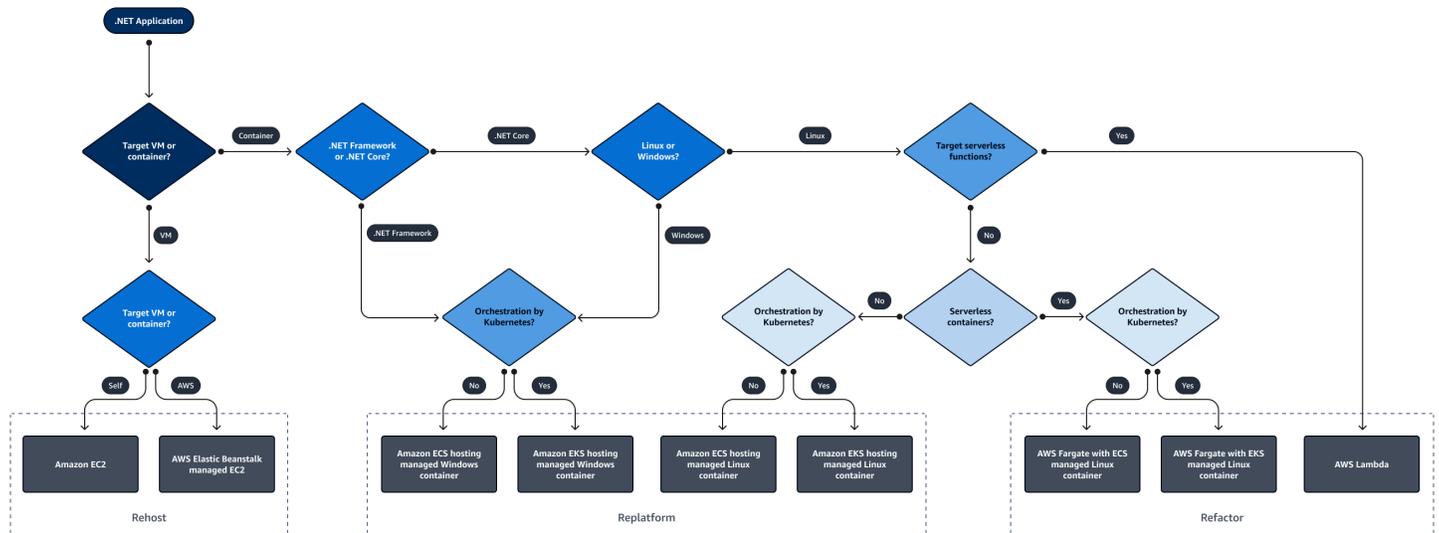
### 使用 Linux 容器而非 Windows

如果您使用 Linux 容器而非 Windows 容器，可以大幅節省成本。例如，如果您在 EC2 Linux 上執行 .NET Core，而不是在 EC2 Windows 上執行 .NET Framework，則可以節省約 45% 的運算成本。如果您使用 ARM 架構 (AWS Graviton) 而非 x86，則可以額外節省 40%。

如果您打算為現有的 .NET Framework 應用程式執行 Linux 型容器，您必須將這些應用程式移植到 .NET (例如 [.NET 6.0](#)) 的現代跨平台版本，才能使用 Linux 容器。與透過降低 Linux 容器成本而節省的成本相比，主要考量是權衡重構的成本。如需將應用程式移植到現代 .NET 的詳細資訊，請參閱 AWS 文件中的 [.NET 的移植助理](#)。

移至現代 .NET (即遠離 .NET Framework) 的另一個好處是，有額外的現代化機會可用。例如，您可以考慮將您的應用程式重新建構為以微服務為基礎的架構，該架構更具可擴展性、敏捷性和成本效益。

下圖說明探索現代化機會的決策程序。



## 利用 Savings Plans

容器可協助您利用 [Compute Savings Plans](#) 來降低 Fargate 成本。彈性折扣模式提供與可轉換預留執行個體相同的折扣。Fargate 定價是根據從您開始下載容器映像到 Amazon ECS 任務終止 (四捨五入至最接近的秒數) 所使用的 vCPU 和記憶體資源。[適用於 Fargate 的 Savings Plans](#) 提供 Fargate 用量高達 50% 的節省，以換取承諾使用特定數量的運算用量 (以每小時美元為單位) 一年或三年。您可以使用 [AWS Cost Explorer](#) 來協助您選擇 Savings Plan。

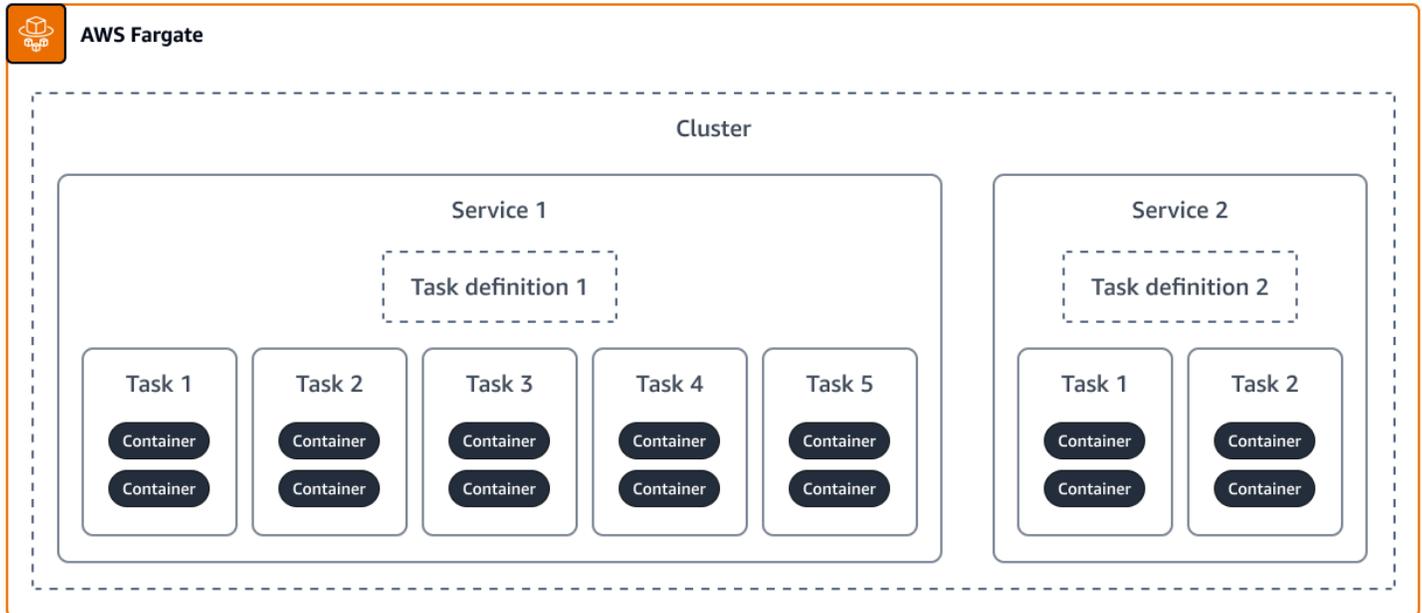
請務必了解 Compute Savings Plans 會套用到讓您最先節省成本的用量。例如，如果您在中執行 t3.medium Linux 執行個體 us-east-2 和相同的 Windows t3.medium 執行個體，Linux 執行個體會先收到 Savings Plan 利益。這是因為 Linux 執行個體具有 50% 的節省潛力，而相同的 Windows 執行個體具有 35% 的節省潛力。如果您的中還有其他 Savings Plan 合格資源正在執行 AWS 帳戶，例如 Amazon EC2 或 Lambda，則不需要先將 Savings Plan 套用到 Fargate。如需詳細資訊，請參閱本指南的 [Savings Plans 文件和 Windows on Amazon EC2 最佳化支出](#) 一節中的 [了解 Savings Plans 如何套用到您的 AWS 用量](#)。Savings Plans [Amazon EC2](#)

## Fargate 任務大小正確

請務必確保 Fargate 任務的大小正確，以達到最高程度的成本最佳化。通常，開發人員在最初決定其應用程式中使用的 Fargate 任務組態時，沒有所有必要的使用資訊。這可能會導致任務過度佈建，然後導致不必要的支出。為了避免這種情況，我們建議您載入在 Fargate 上執行的測試應用程式，以了解特定

任務組態在不同使用案例中如何執行。您可以使用負載測試結果、vCPU、任務的記憶體配置，以及自動擴展政策，在效能和成本之間取得適當的平衡。

下圖顯示 Compute Optimizer 如何針對最佳任務和容器大小產生建議。



其中一種方法是使用負載測試工具，例如 [上的分散式負載測試 AWS](#) 中所述的工具，來建立 vCPU 和記憶體使用率的基準。執行負載測試以模擬一般應用程式負載後，您可以微調任務的 vCPU 和記憶體組態，直到達到基準使用率為止。

## 其他資源

- [Amazon ECS 和 的成本最佳化檢查清單 AWS Fargate](#) (AWS 容器部落格文章 )
- [Amazon ECS 啟動類型的理論成本最佳化：Fargate 與 EC2](#) (AWS 容器部落格文章 )
- [適用於 .NET 的移植助理](#) (AWS 文件 )
- [上的分散式負載測試 AWS](#) (AWS 解決方案程式庫 )
- [AWS Compute Optimizer 在上啟動對 Amazon ECS 服務的支援 AWS Fargate](#) (AWS 雲端財務管理部落格文章 )

# 使用 Graviton 執行個體和容器

## 概觀

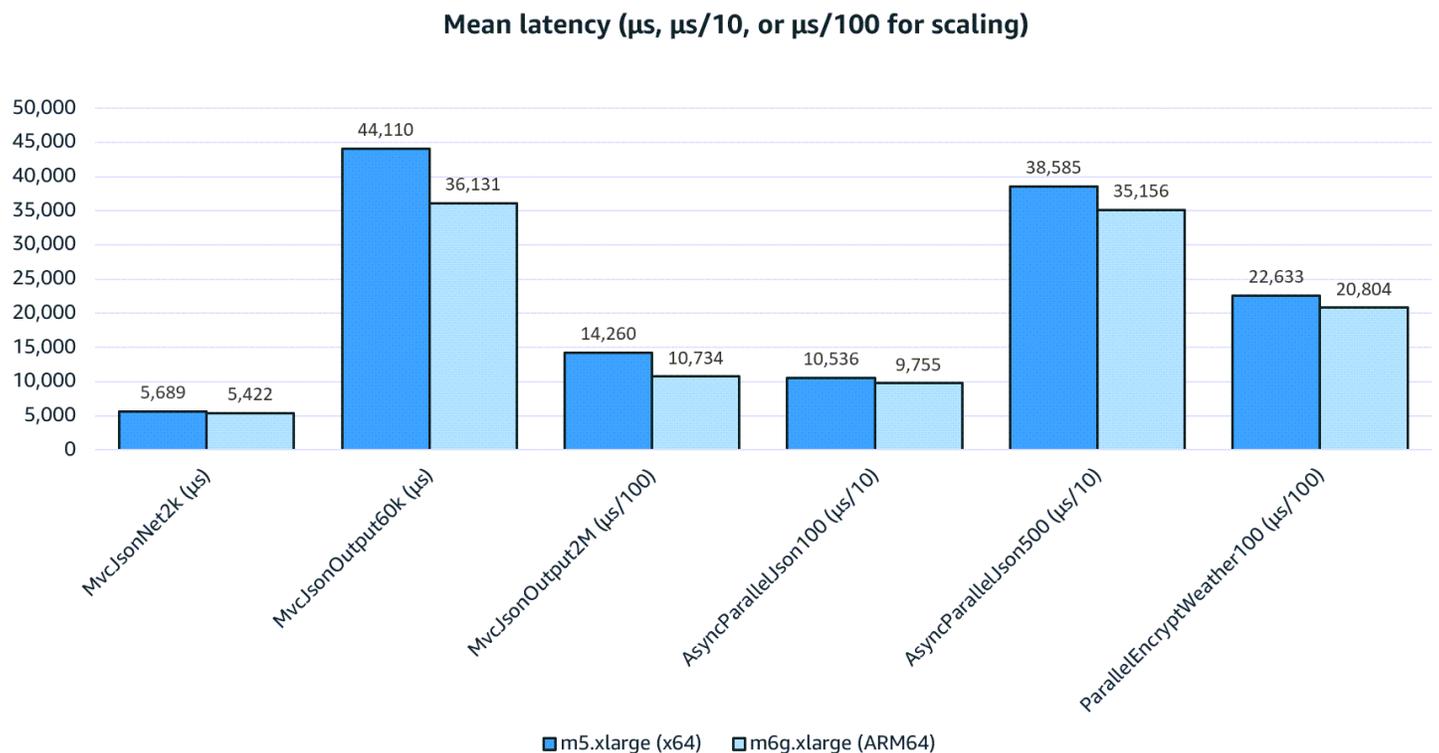
AWS Graviton 執行個體採用設計的 ARM 處理器 AWS，可為在 Amazon Elastic Compute Cloud (Amazon EC2) 中執行的雲端工作負載提供最佳價格效能，包括執行於中的容器 AWS。目前有三代 Graviton 可用於 Amazon EC2。本指南著重於搭配 .NET 應用程式使用 Graviton 2 和 3，因為當您使用最新版本的 Graviton 時，可大幅節省成本。請記住，Graviton 執行個體只會執行 Linux 作業系統。因此，Graviton 執行個體是在 Linux 上執行的 .NET 的強大產品，但不是 Windows 作業系統或舊版 .NET Framework 應用程式的選項。

Graviton 3 比類似的 EC2 執行個體有 60% 的效率，效能提升高達 40%。本指南著重於使用 Graviton 的成本優勢，但請務必注意，Graviton 提供效能改善和環境永續性改善的額外優勢。

## 成本影響

當您切換到 Graviton 時，最多可節省 45%。將任何舊版 .NET Framework 應用程式重構為現代 .NET 版本後，即可解鎖使用 Graviton 執行個體的功能。移至 Graviton 是 .NET 開發人員的有效成本最佳化技術。

下表中的範例顯示遷移至 Graviton 執行個體可實現效能改善的可能性。



如需用於在上圖中建立結果的基準測試方法的完整明細和說明，請參閱 AWS 運算部落格中的[使用 AWS Graviton2：基準測試為 .NET 5 提供支援](#)。

提升效率的其中一個原因是 x86 和 Graviton 之間 vCPU 意義的差異。在 x86 架構中，vCPU 是超執行緒所達成的邏輯核心。在 Graviton 中，vCPU 相當於實體核心，允許 vCPU 完全投入工作負載。

與相當的 x86/x64 執行個體相比，Graviton2 的結果是價格效能提升 40%。Graviton3 在 Graviton2 上提供下列項目：

- 提升效能設定檔，效能提升 25%
- 浮點效能提升高達兩倍
- 高達兩倍的密碼編譯工作負載效能
- 機器學習效能提升高達三倍

此外，Graviton3 是雲端中第一個採用 DDR5 記憶體的执行個體。

下表顯示 Graviton 型執行個體與同等 x86 型執行個體之間節省成本的差異。

此表顯示 Graviton 節省 19.20%。

| 執行個體類型     | 架構  | vCPU | 記憶體 (GB) | 每小時成本 ( 隨需 ) |
|------------|-----|------|----------|--------------|
| t4g.xlarge | ARM | 4    | 16       | 0.1344 美元    |
| t3.xlarge  | x86 | 4    | 16       | 0.1664 美元    |

此表顯示 Graviton 節省 14.99%。

| 執行個體類型      | 架構  | vCPU | 記憶體 (GB) | 每小時成本 ( 隨需 ) |
|-------------|-----|------|----------|--------------|
| c7g.4xlarge | ARM | 16   | 32       | 0.5781 美元    |
| c6i.4xlarge | x86 | 16   | 32       | 0.6800 美元    |

考慮 Graviton 時，請務必測試應用程式的效能描述檔。Graviton 無法取代實體軟體開發實務。您可以使用測試來驗證您是否充分利用基礎運算資源。

## 成本最佳化建議

有多種方式可以利用 Graviton 處理器/執行個體。本節會逐步解說從使用 x86 架構機器移至 Graviton (ARM) 執行個體所需的變更。

### 在 Lambda 中變更執行時間設定

建議您在 中切換執行時間設定 AWS Lambda。如需詳細資訊，請參閱 Lambda 文件中的[修改執行時間環境](#)。由於 .NET 是編譯語言，因此您必須遵循建置程序才能讓此運作。如需如何執行此操作的範例，請參閱 GitHub 中 [Graviton 上的 .NET](#)。

## 容器

針對容器化工作負載，建立多架構容器映像。您可以在 Docker build 命令中指定多個架構來執行此操作。例如：

```
docker buildx build -t "myImageName:latest" --platform linux/amd64,linux/arm64 --push .
```

您也可以使用等工具 AWS Cloud Development Kit (AWS CDK) 來協助[協調建置](#)。如需 Docker 的範例，請參閱《Docker 文件》中的[使用 Docker 桌面建置 Arm 和 x86 的多存檔映像](#)。

## Amazon EC2

若要從 x86/x64 遷移至 ARM，請在編譯步驟中將 ARM 架構設為目標。在 Visual Studio 中，您可以建立 ARM64 CPU。如需說明，請參閱 Microsoft 文件中的[設定專案以鎖定 Arm64 和其他平台](#)。

如果您使用的是 .NET CLI，則在 ARM 機器上執行組建會產生與 Graviton 相容的組建。若要觀看示範，請觀看 [YouTube 上的使用 Arm64 on AWS Graviton2 加速 .NET 6 效能](#)。YouTube 相依性問題將導致編譯時間錯誤，然後可以個別解決。只要有任何相依性的 ARM 程式庫，轉換應該相對簡單。

## 其他資源

- [如何在 Amazon ECS 上使用 Graviton 和 Spot 執行個體建置 ARM 容器並儲存](#) (AWS 部落格)
- [AWS Lambda 採用 AWS Graviton2 處理器的函數 – 在 Arm 上執行函數，並取得高達 34% 的更佳價格效能](#) (AWS 部落格)
- [將 AWS Lambda 函數遷移至 Arm 型 AWS Graviton2 處理器](#) (AWS 部落格)

- [使用 建置並部署 .NET Web 應用程式至 ARM 支援的 AWS Graviton 2 Amazon ECS 叢集 AWS CDK \(AWS 部落格\)](#)
- [Graviton Fast Start – 協助將工作負載移至 AWS Graviton 的新計劃 \(AWS 部落格\)](#)
- [使用 AWS Graviton2 為 .NET 5 提供支援：基準 \(部落格\) AWS](#)

## 支援靜態 .NET Framework 應用程式的動態擴展

### 概觀

將雲端用於應用程式的主要優點之一是彈性，或能夠根據需求擴展運算。這可讓您僅支付所需的運算容量，而不是佈建尖峰用量。網路星期一，其中線上零售商可以快速獲得比平常多出許多倍的流量（例如，[幾分鐘內數千個百分比](#)），是彈性的良好範例。

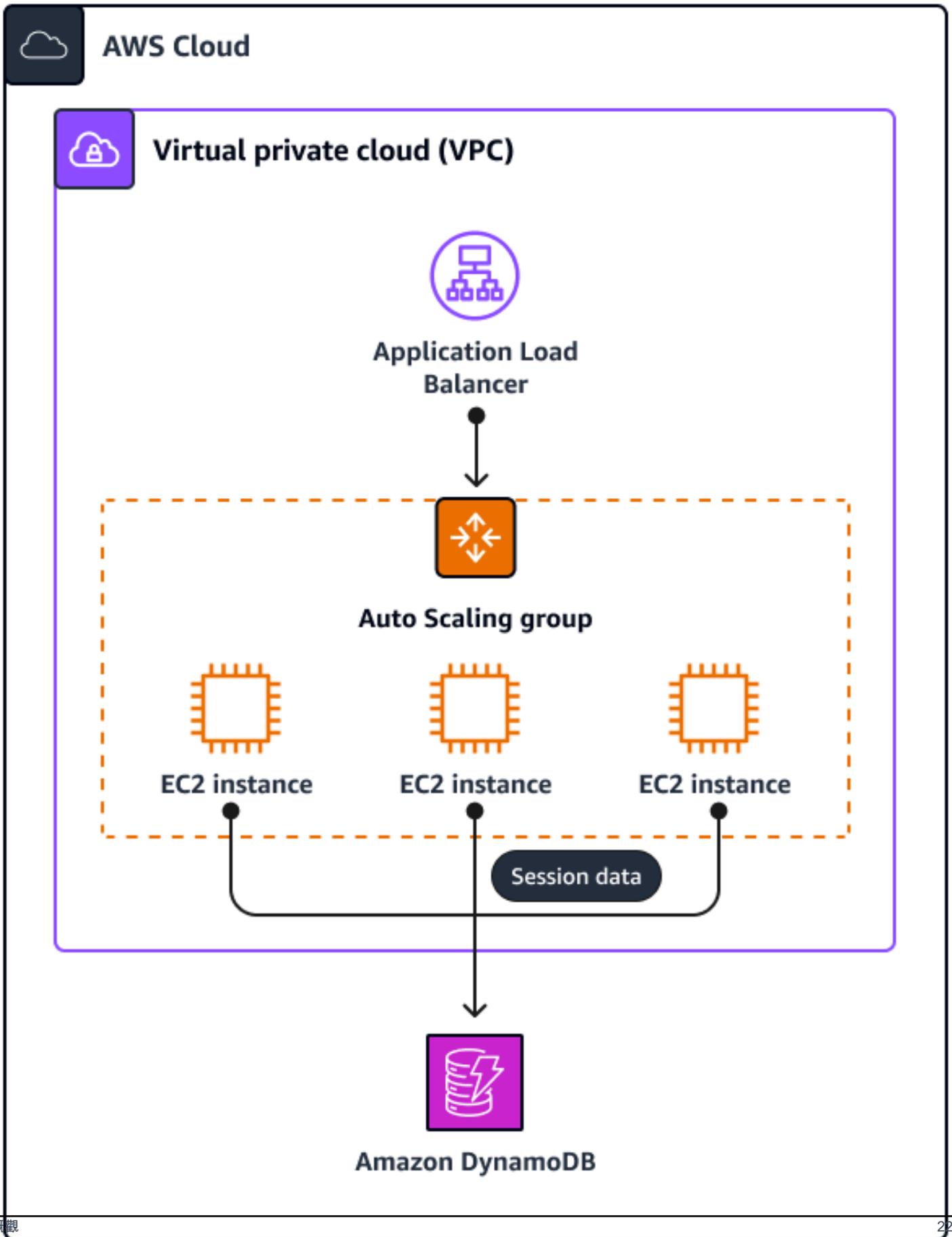
如果您要將舊版 .NET Web 應用程式帶入雲端（例如，在 IIS 上執行的 ASP.NET Framework 應用程式），由於應用程式的狀態性質，快速擴展負載平衡伺服器陣列的能力可能很困難或不可能。使用者工作階段資料會存放在應用程式的記憶體中，通常具有 [ASP.NET 工作階段狀態](#) 或靜態變數，這些變數包含必須保留的跨請求資料。使用者工作階段親和性通常透過負載平衡器黏性工作階段來維護。

這證明在操作上具有挑戰性。需要增加容量時，您必須刻意佈建和新增伺服器。這可能是緩慢的程序。在修補或意外失敗的情況下停用節點可能會對最終使用者體驗造成問題，因此與受影響節點相關聯的所有使用者都會失去狀態。最好是，這需要使用者再次登入。

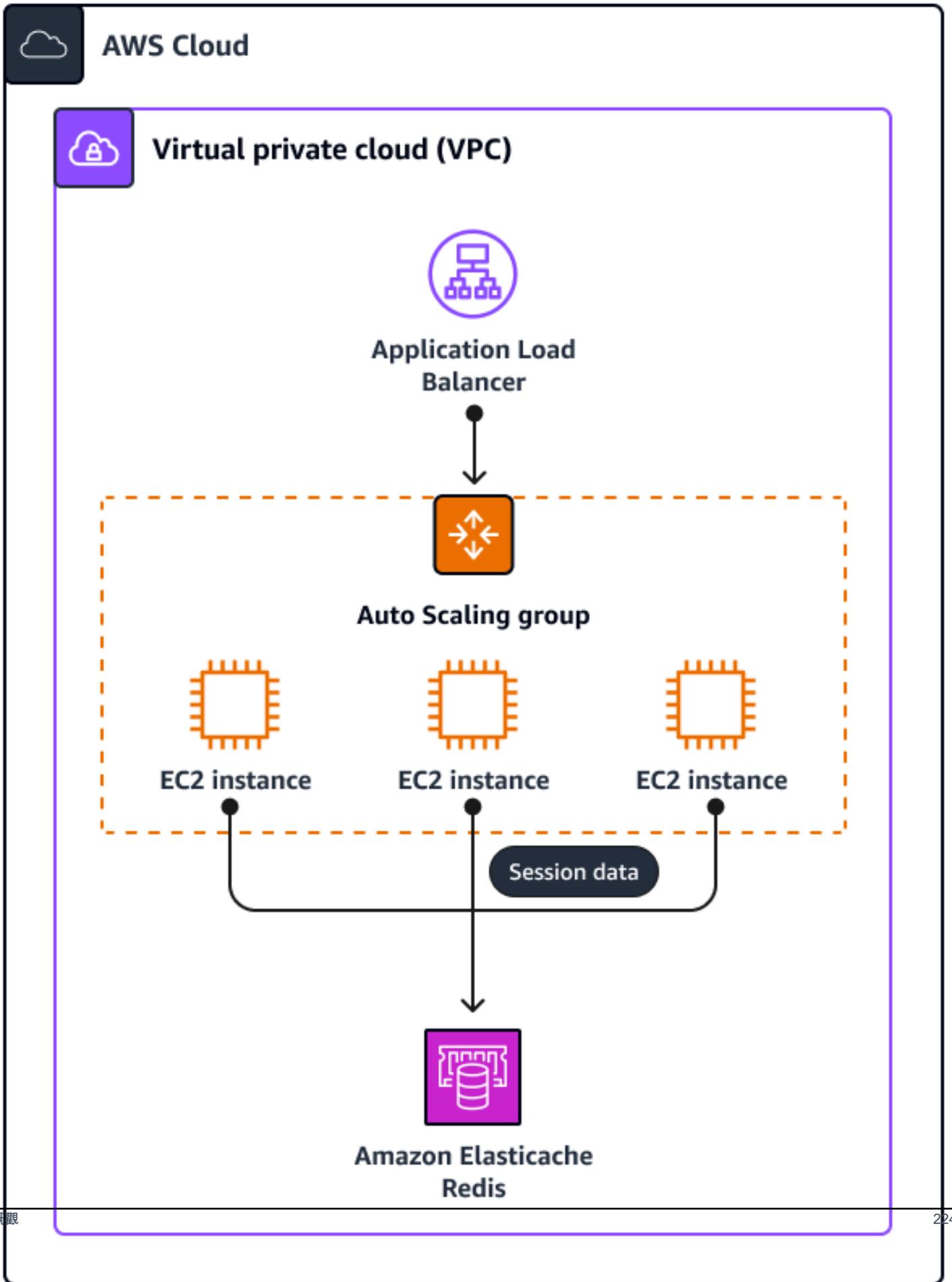
透過集中 ASP.NET 應用程式的工作階段狀態，並將自動調整規模規則套用至舊版 ASP.NET 應用程式，您可以利用雲端的彈性，並可能在執行應用程式時節省成本。例如，您可以透過運算可擴展性降低成本，但您也可以從可用的不同定價模型中進行選擇，例如減少 [預留執行個體用量](#) 和使用 [Amazon Spot 執行個體定價](#)。

兩種常見的技術包括使用 [Amazon DynamoDB 做為工作階段狀態提供者](#)，以及使用 [Amazon ElastiCache \(Redis OSS\) 做為 ASP.NET 工作階段存放區](#)。

下圖顯示使用 DynamoDB 做為工作階段狀態提供者的架構。



下圖顯示使用 ElastiCache (Redis OSS) 做為工作階段狀態提供者的架構。



## 成本影響

若要判斷生產應用程式擴展的優點，建議您建立實際需求的模型。本節會做出下列假設來建立範例應用程式的模型：

- 從輪換中新增和移除的執行個體是相同的，不會引入任何執行個體大小變化。
- 伺服器使用率永遠不會低於兩個作用中伺服器，以維持應用程式的高可用性。
- 伺服器數量會隨流量線性擴展（也就是說，兩倍的流量需要兩倍的運算）。
- 流量會在一個月內以六小時為單位遞增建模，一天內的變化和 10 倍流量一天的異常流量峰值（例如，促銷銷售）。週末流量以基本使用率建模。
- 夜間流量以基本使用率建模，而工作日流量以 4 倍使用率建模。
- 預留執行個體定價使用一年無預付款定價。一般日間定價使用隨需定價，而爆量需求使用 Spot 執行個體定價。

下圖說明此模型如何利用 .NET 應用程式的彈性，而不是佈建尖峰用量。這可節省約 68%。

Comparison of cumulative costs for peak provisioning and autoscaling



如果您使用 DynamoDB 做為工作階段狀態儲存機制，請使用下列參數：

```
Storage: 20GB
Session Reads: 40 million
Session Writes: 20 million
Pricing Model: On demand
```

此服務的估計每月成本約為 35.00 美元。

如果您使用 ElastiCache (Redis OSS) 做為工作階段狀態儲存機制，請使用下列參數：

```
Number of Nodes: 3
Node size: cache.t4g.medium
Pricing Model: 1y reserved
```

此服務的估計每月成本約為 91.00 美元。

## 成本最佳化建議

第一步是在舊版 .NET 應用程式中實作工作階段狀態。如果您使用 ElastiCache 做為狀態儲存機制，請遵循 AWS 開發人員工具部落格中 [ElastiCache 作為 ASP.NET 工作階段存放區](#) 的指導。如果您使用的是 DynamoDB，請遵循適用於 .NET 的 SDK 文件中 [什麼是適用於 .NET 的 AWS SDK](#) 的指引。

如果應用程式使用 InProc 工作階段開頭，請確定您計劃存放在工作階段中的所有物件都可以序列化。若要這樣做，請使用 `SerializableAttribute` 屬性來裝飾類別，其執行個體將存放在工作階段中。例如：

```
[Serializable()]
public class TestSimpleObject {
    public string SessionProperty {get;set;}
}
```

此外，所有使用中的伺服器之間的 .NET MachineKey 必須相同。這通常是從常見 Amazon Machine Image (AMI) 建立執行個體的情況。例如：

```
<machineKey
validationKey="some long hashed value"
decryptionKey="another long hashed value"
validation="SHA1"/>
```

不過，請務必確保如果基礎映像變更，則會使用相同的 .NET 機器映像進行設定（可在 IIS 或伺服器層級設定）。如需詳細資訊，請參閱 Microsoft 文件中的 [SystemWebSectionGroup.MachineKey 屬性](#)。

最後，您必須判斷將伺服器新增至 Auto Scaling 群組以回應擴展事件的機制。有幾種方法可以完成此操作。我們建議您使用下列方法來將 .NET Framework 應用程式無縫部署到 Auto Scaling 群組中的 EC2 執行個體：

- 使用 [EC2 Image Builder](#) 設定包含完整設定伺服器 and 應用程式的 AMI。然後，您可以使用此 AMI 來設定 [Auto Scaling 群組的啟動範本](#)。
- 使用 [AWS CodeDeploy](#) 部署您的應用程式。CodeDeploy 可直接與 [Amazon EC2 Auto Scaling](#) 整合。這提供了為每個應用程式版本建立新的 AMI 的替代方案。

## 其他資源

- [使用 EC2 Image Builder 建立映像](#) (EC2 Image Builder 文件 )
- [使用 AWS CodeDeploy 搭配 Visual Studio Team Services 部署 .NET Web 應用程式](#) (AWS 開發人員工具部落格 )

## 使用快取來減少資料庫需求

### 概觀

您可以使用快取作為有效的策略，以協助降低 .NET 應用程式的成本。當應用程式需要頻繁存取資料時，許多應用程式會使用後端資料庫，例如 SQL Server。維護這些後端服務以因應需求的成本可能很高，但您可以使用有效的快取策略，藉由減少大小和擴展需求來減少後端資料庫的負載。這可協助您降低成本並改善應用程式的效能。

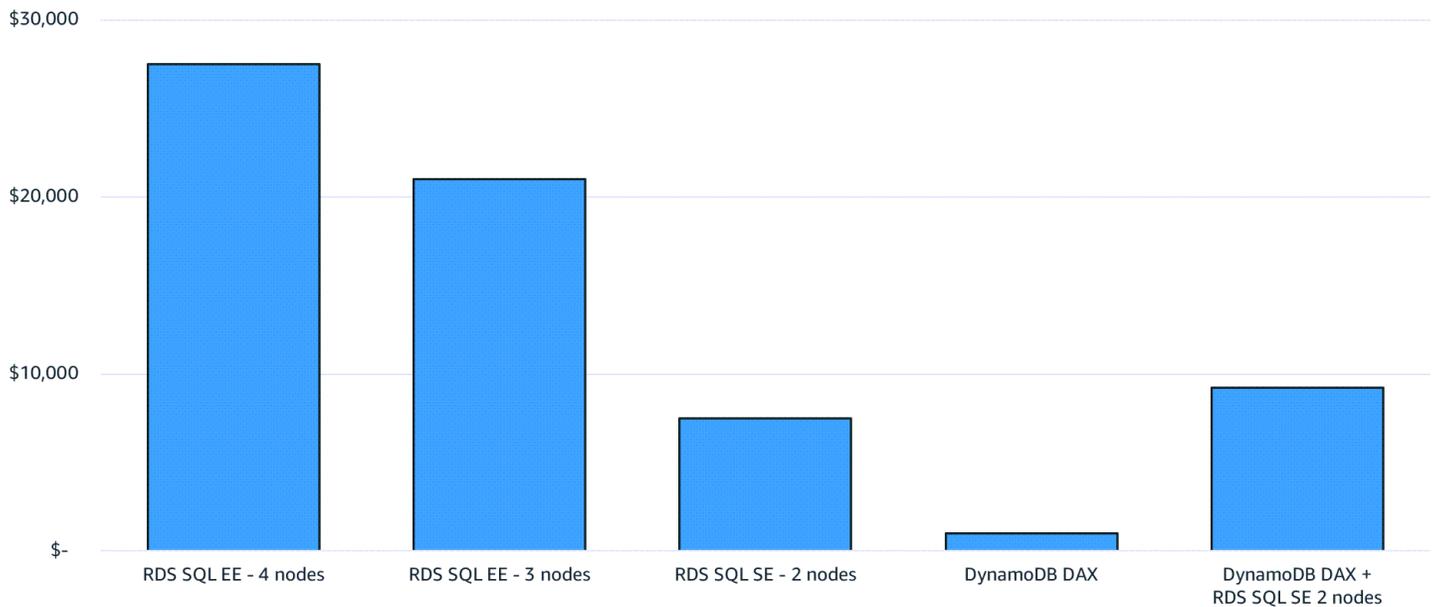
快取是一種有用的技術，可節省與讀取繁重工作負載相關的成本，這些工作負載使用更昂貴的資源，例如 SQL Server。請務必為您的工作負載使用正確的技術。例如，本機快取無法擴展，需要您維護應用程式每個執行個體的本機快取。您應該權衡與潛在成本相比的效能影響，以便基礎資料來源的成本較低會抵銷與快取機制相關的任何額外成本。

### 成本影響

SQL Server 需要您在調整資料庫大小時考慮讀取請求。這可能會影響成本，因為您可能需要引入僅供讀取複本以適應負載。如果您使用的是僅供讀取複本，請務必了解這些複本僅適用於 SQL Server Enterprise Edition。此版本需要比 SQL Server Standard Edition 更昂貴的授權。

下圖旨在協助您了解快取有效性。它顯示具有四個執行 SQL Server Enterprise Edition 的 db.m4.2xlarge 節點的 Amazon RDS for SQL Server。它部署在具有一個僅供讀取複本的多可用區組態中。專屬讀取流量（例如 SELECT 查詢）會導向僅供讀取複本。相比之下，Amazon DynamoDB 使用 r4.2xlarge 兩個節點 DynamoDB Accelerator (DAX) 叢集。

下圖顯示移除對處理高讀取流量之專用僅供讀取複本的需求的結果。



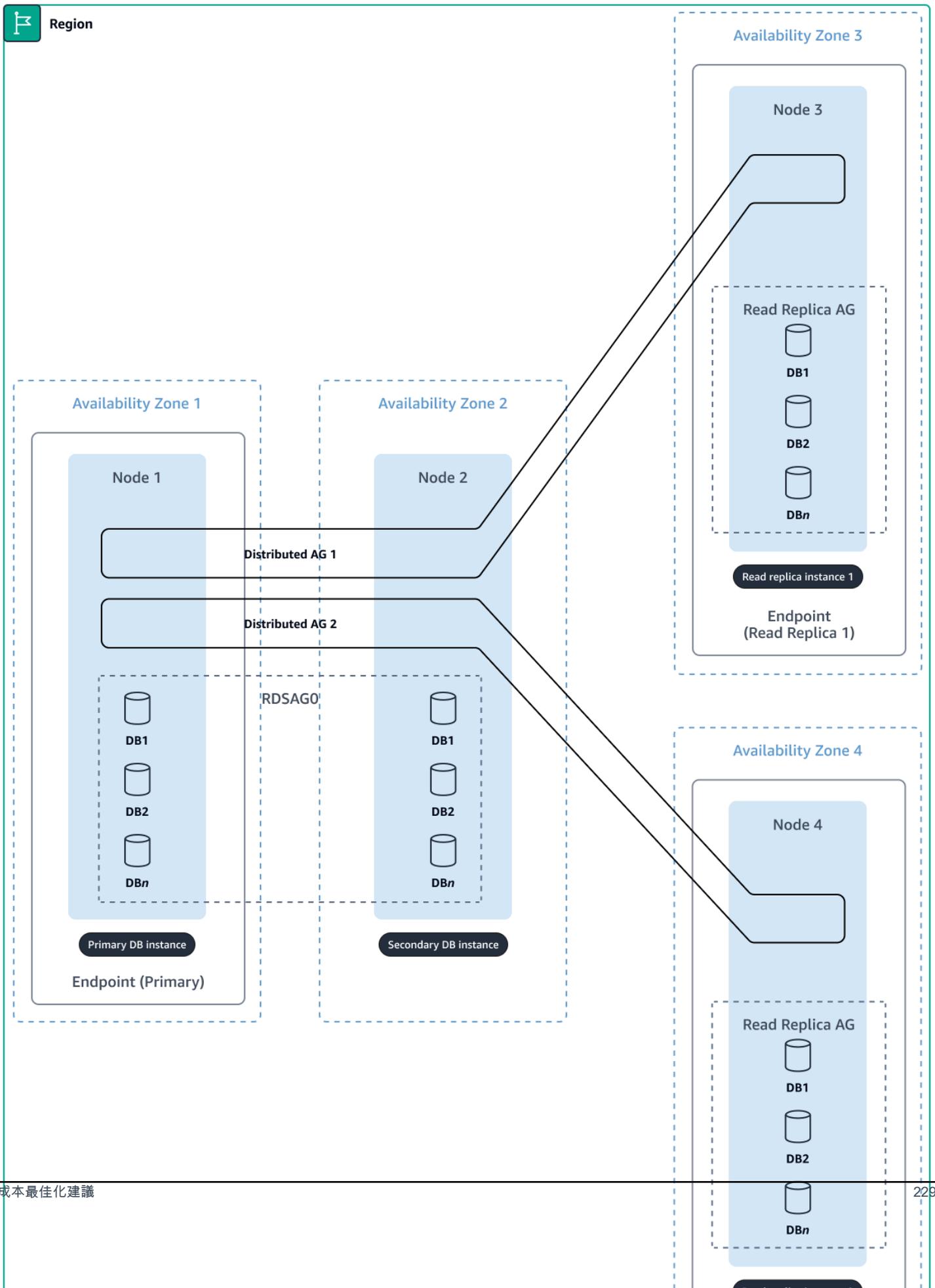
您可以將本機快取與 Amazon RDS 上的 SQL Server 並排引入 DAX 做為快取層，藉此大幅節省成本。此層會從 SQL Server 卸載，並減少執行資料庫所需的 SQL Server 大小。

## 成本最佳化建議

### 本機快取

本機快取是快取託管在內部部署環境或雲端中應用程式內容的最常見方式之一。這是因為實作相對簡單且直覺式。本機快取涉及從資料庫或其他來源取得內容，以及在記憶體或磁碟本機快取，以便更快速存取。這種方法雖然易於實作，但不適用於某些使用案例。例如，這包括快取內容需要隨著時間保留的使用案例，例如保留應用程式狀態或使用者狀態。另一個使用案例是，需要從其他應用程式執行個體存取快取的內容。

下圖說明具有四個節點和兩個僅供讀取複本的高可用性 SQL Server 叢集。



使用本機快取時，您可能需要在多個 EC2 執行個體之間負載平衡流量。每個執行個體都必須維護自己的本機快取。如果快取存放有狀態的資訊，則需要定期遞交至資料庫，而且使用者可能需要針對每個後續請求（黏性工作階段）轉送至相同的執行個體。這在嘗試擴展應用程式時存在挑戰，因為某些執行個體可能會過度使用，而有些執行個體由於流量分佈不均而未充分利用。

對於 .NET 應用程式，您可以使用本機快取，無論是記憶體內或使用本機儲存。若要這樣做，您可以新增功能，將物件存放在磁碟上，並在需要時擷取它們，或查詢資料庫中的資料並將其保留在記憶體中。若要在 C# 中從 SQL Server 本機儲存資料時執行本機快取，例如，您可以使用 MemoryCache 和 LiteDB 程式庫的組合。MemoryCache 提供記憶體內快取，而 LiteDB 是快速輕量型的內嵌 NoSQL 磁碟型資料庫。

若要執行記憶體內快取，請使用 .NET Library System.Runtime.MemoryCache。下列程式碼範例示範如何使用 System.Runtime.Caching.MemoryCache 類別來快取記憶體中的資料。此類別可讓您暫時將資料存放在應用程式的記憶體中。這可以透過減少從資料庫或 API 等更昂貴的資源擷取資料的需求，協助改善應用程式的效能。

以下是程式碼的運作方式：

1. `_memoryCache` 建立 MemoryCache 名為 `dataCache` 的私有靜態執行個體。快取會指定名稱 (`dataCache`) 來識別它。然後，快取會存放和擷取資料。
2. `GetData` 方法是一種採用兩個引數的一般方法：`string` 金鑰和名為 `Func<T>` 委派 `getData`。金鑰用於識別快取的資料，而 `getData` 委派代表在資料不存在於快取時執行的資料擷取邏輯。
3. 方法會先使用 `_memoryCache.Contains(key)` 方法檢查快取中是否存在資料。如果資料位於快取中，則方法會使用 `_memoryCache.Get(key)` 並將其轉換為預期的類型 `T`。
4. 如果資料不在快取中，方法會呼叫 `getData` 委派人來擷取資料。然後，它會使用 `_memoryCache.Add(key, data, DateTimeOffset.Now.AddMinutes(10))` 將資料新增至快取。此呼叫指定快取項目應該會在 10 分鐘後過期，此時資料會自動從快取中移除。
5. `ClearCache` 方法會採用 `string` 金鑰做為引數，並使用 `_memoryCache.Remove(key)` 從快取中移除與該金鑰相關聯的資料。

```
using System;
using System.Runtime.Caching;

public class InMemoryCache
{
    private static MemoryCache _memoryCache = new MemoryCache("dataCache");

    public static T GetData<T>(string key, Func<T> getData)
```

```
{
    if (_memoryCache.Contains(key))
    {
        return (T)_memoryCache.Get(key);
    }

    T data = getData();
    _memoryCache.Add(key, data, DateTimeOffset.Now.AddMinutes(10));

    return data;
}

public static void ClearCache(string key)
{
    _memoryCache.Remove(key);
}
}
```

您可以使用下列程式碼：

```
public class Program
{
    public static void Main()
    {
        string cacheKey = "sample_data";

        Func<string> getSampleData = () =>
        {
            // Replace this with your data retrieval logic
            return "Sample data";
        };

        string data = InMemoryCache.GetData(cacheKey, getSampleData);
        Console.WriteLine("Data: " + data);
    }
}
```

下列範例示範如何使用 [LiteDB](#) 快取本機儲存中的資料。您可以使用 LiteDB 做為記憶體內快取的替代方案或補充。下列程式碼示範如何使用 LiteDB 程式庫快取本機儲存中的資料。LocalStorageCache 類別包含管理快取的主要函數。

```
using System;
using LiteDB;
```

```
public class LocalStorageCache
{
    private static string _liteDbPath = @"Filename=LocalCache.db";

    public static T GetData<T>(string key, Func<T> getData)
    {
        using (var db = new LiteDatabase(_liteDbPath))
        {
            var collection = db.GetCollection<T>("cache");
            var item = collection.FindOne(Query.EQ("_id", key));

            if (item != null)
            {
                return item;
            }
        }

        T data = getData();

        using (var db = new LiteDatabase(_liteDbPath))
        {
            var collection = db.GetCollection<T>("cache");
            collection.Upsert(new BsonValue(key), data);
        }

        return data;
    }

    public static void ClearCache(string key)
    {
        using (var db = new LiteDatabase(_liteDbPath))
        {
            var collection = db.GetCollection("cache");
            collection.Delete(key);
        }
    }
}

public class Program
{
    public static void Main()
    {
        string cacheKey = "sample_data";
    }
}
```

```
Func<string> getSampleData = () =>
{
    // Replace this with your data retrieval logic
    return "Sample data";
};

string data = LocalStorageCache.GetData(cacheKey, getSampleData);
Console.WriteLine("Data: " + data);
}
}
```

如果您有靜態快取或靜態檔案不常變更，您也可以將這些檔案存放在 Amazon Simple Storage Service (Amazon S3) 物件儲存體中。應用程式可以在啟動時擷取靜態快取檔案，以便在本機使用。如需如何使用 .NET 從 Amazon S3 擷取檔案的詳細資訊，請參閱 Amazon S3 文件中的[下載物件](#)。

## 使用 DAX 快取

您可以使用可在所有應用程式執行個體之間共用的快取層。[DynamoDB Accelerator \(DAX\)](#) 是 DynamoDB 的全受管、高度可用的記憶體內快取，可提供十倍的效能改善。您可以使用 DAX 來降低成本，方法是減少在 DynamoDB 資料表中過度佈建讀取容量單位的需求。這對於讀取繁重且需要對個別金鑰重複讀取的工作負載特別有用。

DynamoDB 是隨需定價，或使用佈建容量定價，因此每月的讀取和寫入次數對成本有所貢獻。如果您有讀取繁重的工作負載，DAX 叢集可以透過減少 DynamoDB 資料表上的讀取數量來協助降低成本。如需如何設定 DAX 的說明，請參閱 [DynamoDB 文件中的使用 DynamoDB Accelerator \(DAX\) 進行記憶體內加速](#)。DynamoDB 如需有關 .NET 應用程式整合的資訊，請觀看在 YouTube 上[將 Amazon DynamoDB DAX 整合到您的 ASP.NET 應用程式](#)。

## 其他資源

- [使用 DynamoDB Accelerator \(DAX\) 的記憶體內加速 - Amazon DynamoDB](#) (DynamoDB 文件)
- [將 Amazon DynamoDB DAX 整合到您的 ASP.NET 應用程式](#) (YouTube)
- [下載物件](#) (Amazon S3 文件)

# 考慮無伺服器 .NET

## 概觀

無伺服器運算已成為建置和部署應用程式的熱門方法。主要是因為無伺服器方法在建置現代架構時所提供的可擴展性和敏捷性。不過，在某些情況下，考慮無伺服器運算的成本影響非常重要。

Lambda 是一種無伺服器運算平台，可讓開發人員執程式碼，而不需要專用伺服器。對於尋求降低基礎設施成本的 .NET 開發人員而言，Lambda 是特別有吸引力的選項。使用 Lambda，.NET 開發人員可以開發和部署具有高度可擴展性和潛在成本效益的應用程式。透過使用無伺服器方法，開發人員不再佈建伺服器來處理應用程式請求。反之，開發人員可以建立隨需執行的函數。這使得無伺服器方法比執行、管理和擴展虛擬機器更具可擴展性、可管理性，而且可能更具成本效益。因此，您只需支付應用程式使用的資源，而不必擔心資源使用率不足或伺服器維護成本。

開發人員可以使用現代的跨平台 .NET 版本來建置快速、有效率且符合成本效益的無伺服器應用程式。 .NET Core 和更新版本是免費且開放原始碼的架構，比先前的 .NET Framework 版本更適合在無伺服器平台上執行。這可讓開發人員縮短開發時間並提高應用程式效能。現代 .NET 也支援多種程式設計語言，包括 C# 和 F#。因此，對於希望在雲端中建置現代架構的開發人員來說，這是一個吸引人的選項。

本節說明如何使用 Lambda 做為無伺服器選項來節省成本。您可以透過微調 Lambda 函數的執行設定檔、調整 Lambda 函數的記憶體配置大小、使用[原生 AOT](#)，以及移至 Graviton 型函數，進一步最佳化成本。

## 成本影響

您可以降低成本的程度取決於幾個因素，包括無伺服器函數除了配置的記憶體數量和每個函數的持續時間之外，還會執行多少次執行。AWS Lambda 提供免費方案，其中包括每月 100 萬次免費請求和每月 400,000 GB 的運算時間。您可以大幅降低這些免費方案限制或接近這些限制之工作負載的每月成本。

使用負載平衡器搭配 Lambda 函數做為目標時，可能還需要支付額外費用。這計算方式為負載平衡器針對[Lambda 目標](#)處理的資料量。

## 成本最佳化建議

### 正確調整 Lambda 函數的大小

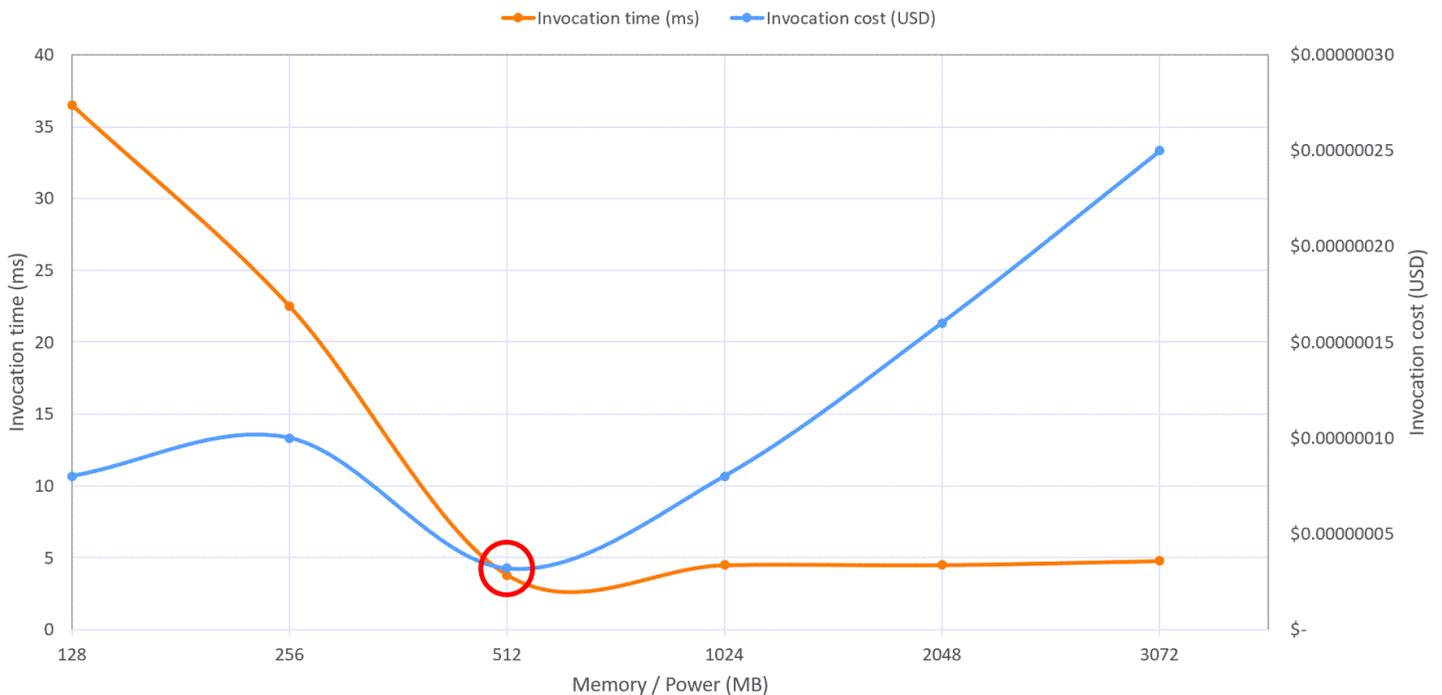
適當調整大小是 .NET 型 Lambda 函數中成本最佳化的基本實務。此程序涉及識別最佳記憶體組態，在效能與成本效益之間取得平衡，而不需要變更程式碼。

透過設定 Lambda 函數的記憶體，範圍從 128 MB 到 10,240 MB，您也可以調整調用期間可用的 vCPU 數量。這可讓記憶體或 CPU 繫結的應用程式在執行期間存取其他資源，進而降低調用持續時間和整體成本。

不過，識別以 .NET 為基礎的 Lambda 函數的最佳組態可以是手動且耗時的程序，特別是在頻繁變更的情況下。[AWS Lambda Power Tuning 工具](#)可以協助您根據範例承載分析一組記憶體組態，以識別適當的組態。

例如，增加以 .NET 為基礎的 Lambda 函數的記憶體可以改善總叫用時間和降低成本，而不會影響效能。函數的最佳記憶體組態可能有所不同。AWS Lambda Power Tuning 工具可協助識別每個函數最具成本效益的組態。

在下列範例圖表中，總調用時間會隨著此 Lambda 函數的記憶體增加而改善。這會導致總執行的成本降低，而不會影響函數的原始效能。對於此函數，函數的最佳記憶體組態為 512 MB，因為這是資源使用率對於每次調用的總成本最有效率的地方。這因函數而異，而且在 Lambda 函數上使用工具可以識別它們是否受益於正確的大小。



我們建議您定期完成此練習，作為發佈新更新時任何整合測試的一部分。如果不常更新，請定期執行此練習，以確保函數已調整且大小正確。識別 Lambda 函數的適當記憶體設定後，您可以為程序新增適當大小。AWS Lambda Power Tuning 工具會產生程式設計輸出，供您的 CI/CD 工作流程在發行新程式碼期間使用。這可讓您自動化記憶體組態。

您可以免費下載 [AWS Lambda Power Tuning 工具](#)。如需如何使用工具的指示，請參閱 [如何在 GitHub 中執行狀態機器](#)。GitHub

Lambda 也支援原生 AOT，這可讓 .NET 應用程式預先編譯。這可以透過減少 .NET 函數的執行時間來協助降低成本。如需建立原生 AOT 函數的詳細資訊，請參閱 Lambda 文件中的 [具有原生 AOT 編譯的 .NET 函數](#)。

## 避免閒置等待時間

Lambda 函數持續時間是用於計算帳單的一個維度。當函數程式碼發出封鎖呼叫時，您需支付等待接收回應的時間費用。當 Lambda 函數鏈結在一起，或函數充當其他函數的協調器時，此等待時間可能會增加。如果您有批次操作或訂單交付系統等工作流程，這會增加管理開銷。此外，可能無法在最長 15 分鐘的 Lambda 逾時內完成所有工作流程邏輯和錯誤處理。

建議您重新建構解決方案以 [AWS Step Functions](#) 用作工作流程的協調器，而不是在函數程式碼中處理此邏輯。使用標準工作流程時，您需要支付工作流程內每個 [狀態轉換](#) 的費用，而非工作流程的總持續時間。此外，您可以將重試、等待條件、錯誤工作流程和 [回呼](#) 的支援移至狀態條件，以允許 Lambda 函數專注於商業邏輯。如需詳細資訊，請參閱 AWS 運算部落格中的 [最佳化 AWS Lambda 成本 – 第 2 部分](#)。

## 移至以 Graviton 為基礎的函數

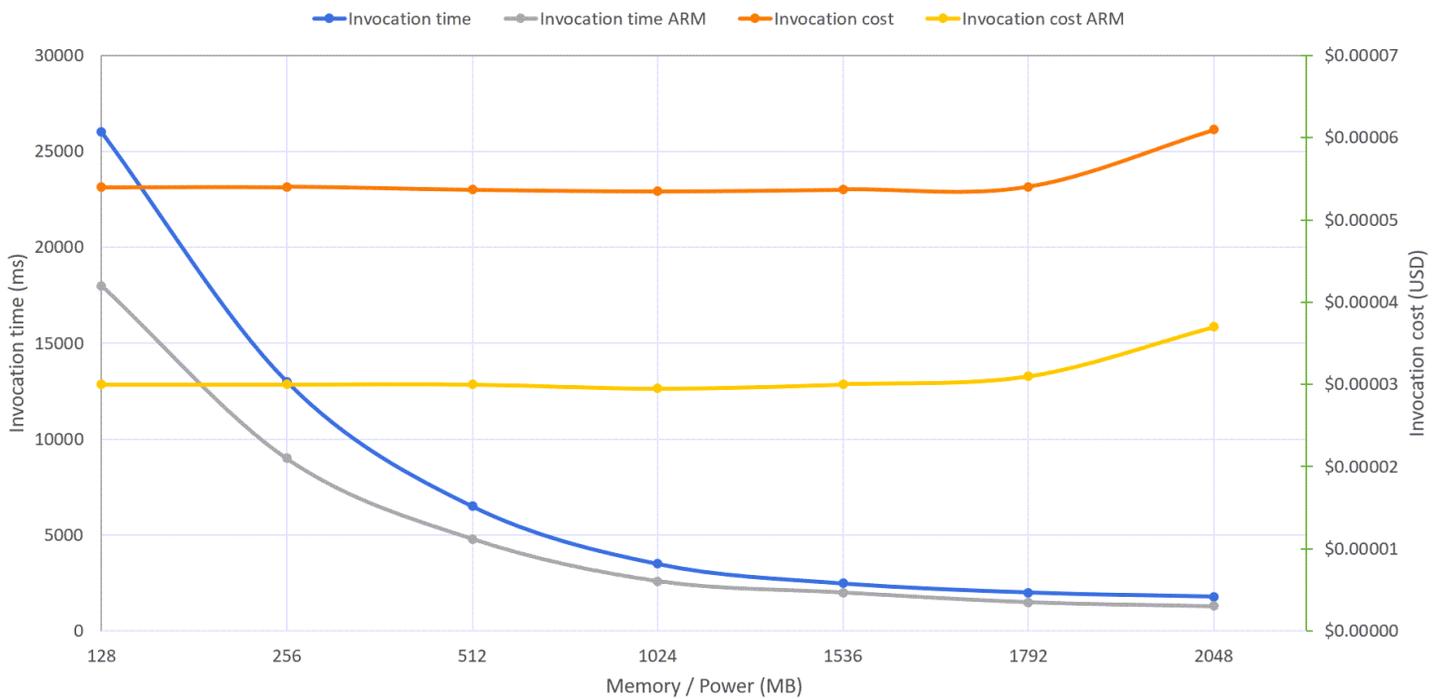
採用新一代 Graviton2 處理器的 Lambda 函數現已正式推出。Graviton2 函數使用 ARM 型處理器架構，旨在為各種無伺服器工作負載提供高達 19% 的更佳效能，成本降低 20%。採用 Graviton2 處理器的函數具有更低的延遲和更好的效能，非常適合為關鍵任務無伺服器應用程式提供支援。

遷移至 Graviton 型 Lambda 函數對於希望最佳化其 Lambda 成本的 .NET 開發人員來說，是經濟實惠的選項。Graviton 型函數使用 ARM 型處理器，而非傳統的 x86 處理器。這可以大幅節省成本，而不會犧牲效能。

雖然移至 Graviton 型函數有幾個優點，但我們建議您考慮幾個挑戰和考量事項。例如，Graviton 型函數需要使用 Amazon Linux 2，這可能無法與所有 .NET 應用程式相容。此外，第三方程式庫或相依性可能有與 ARM 型處理器不相容的相容性問題。

如果您正在執行 .NET Framework 應用程式，並想要利用無伺服器搭配 Lambda，您可以考慮使用適用於 .NET 的 [移植助理](#)，將應用程式移植到現代 .NET。這可協助您加速將舊版 .NET 應用程式移轉至現代 .NET，讓應用程式能夠在 Linux 上執行。

下圖比較 x86 和 ARM/Graviton2 架構結果，用於運算質數的函數。



函數使用單一執行緒。當記憶體設定為 1.8 GB 時，會報告這兩個架構的最低持續時間。除此之外，Lambda 函數可以存取超過 1 個 vCPU，但在此情況下，函數無法使用額外的電源。基於相同原因，成本在記憶體高達 1.8 GB 的情況下是穩定的。隨著記憶體的增加，成本會增加，因為此工作負載沒有額外的效能優勢。Graviton2 處理器明確為這個運算密集型函數提供更好的效能並降低成本。

若要將函數設定為搭配 Graviton 使用和 ARM 型處理器，請執行下列動作：

1. 登入 [AWS Management Console](#) 然後開啟 [Lambda 主控台](#)。
2. 選擇 Create function (建立函數)。
3. 針對 Function name (函數名稱)，輸入名稱。
4. 針對執行時間，選擇 .NET 6 (C#/PowerShell)。
5. 針對架構，選取 arm64。
6. 進行您需要的任何其他組態，然後選擇建立函數。

## 其他資源

- [Lambda 函數做為目標](#) (AWS 文件)
- [使用最佳化 AWS Lambda 成本和效能 AWS Compute Optimizer](#) (AWS 運算部落格)
- [最佳化 AWS Lambda 成本 – 第 1 部分](#) (AWS 運算部落格)
- [最佳化 AWS Lambda 成本 – 第 2 部分](#) (AWS 運算部落格)

- [AWS Lambda 使用 .NET 7 在上建置無伺服器 .NET 應用程式 \(AWS 運算部落格\)](#)

## 考慮專門建置的資料庫

### 概觀

執行 Microsoft 型工作負載最昂貴的層面之一，來自於商業資料庫的授權，例如 SQL Server。企業通常會在 SQL Server 上標準化為選擇的資料庫平台，並融入組織的開發文化。無論使用案例為何，開發人員通常會選擇關聯式 SQL Server 型模型。原因包括：

- 企業已有可用的 SQL Server 執行個體和/或授權。
- 團隊已透過使用共用程式庫、ORMs 和商業邏輯，習慣 SQL 程式設計模型。
- 管理層不知道替代方案。
- 開發人員不知道替代方案。

專門建置的資料庫可以容納使用案例的資料存取模式。這些資料庫隨著採用更現代化的架構（例如微服務）以及個別應用程式的範圍縮小而逐漸受到企業採用。

正在專門建置的資料庫不會排除關聯式模型，或需要 NoSQL（非關聯式）模型。事實上，在選擇關聯式資料庫以回應工作負載的特定需求時，關聯式資料庫會被視為專門建置。使用專門建置的資料庫可協助團隊降低與其 .NET 應用程式相關的資料庫成本，同時獲得標準雲端優勢，例如可擴展性、彈性，以及減少無差別的繁重作業。

下表顯示提供的專用資料庫 AWS。

| 資料庫   | Type | 特性   |
|---|------|--|
| <a href="#">Amazon Aurora PostgreSQL</a><br>或 <a href="#">Amazon Aurora MySQL</a> | 關聯式  | 資料具有固定結構的使用案例<br><br>關聯式資料庫會透過 ACID 交易自然維持資料一致性  |
| <a href="#">Amazon DynamoDB</a>   | 鍵/值對 | 使用雜湊資料表資料結構存放資料的 NoSQL 資料庫<br><br>高效能儲存和擷取非結構化資料 |

| 資料庫                                | Type    | 特性   |
|------------------------------------|---------|--|
|                                    |         | 使用案例包括使用者設定檔、工作階段狀態和購物車資料  |
| <a href="#">Amazon ElastiCache</a> | 記憶體內    | <p>高效能 NoSQL 資料庫，可將非結構化資料存放在存取時間低於毫秒的記憶體中</p> <p>用於經常存取的暫時性資料，例如使用者工作階段，以及作為在其他較慢的資料存放區前面的快取層</p> <p>包括對 ElastiCache (Redis OSS) 和 ElastiCache (Memcached) 的支援</p> |
| <a href="#">Amazon MemoryDB</a>    | 耐用的記憶體內 | Redis 相容的專用資料庫，具有耐用的儲存   |
| <a href="#">Amazon Timestream</a>  | 時間序列    | <p>專為暫時順序高輸送量資料擷取而設計的資料庫</p> <p>使用案例包括物聯網 (IoT) 應用程式和儲存指標或遙測資料</p>   |
| <a href="#">Amazon DocumentDB</a>  | 文件      | <p>NoSQL 資料庫，可在沒有規定結構或強制與其他資料的關係的情況下存放資料</p> <p>通常用於讀取密集型工作負載，例如產品目錄</p>   |
| <a href="#">Amazon Neptune</a>     | 圖表      | <p>同時保留資料和資料項目之間連線表示的 NoSQL 資料庫</p> <p>使用案例包括詐騙偵測、建議引擎和社交應用程式</p>  |

| 資料庫                              | Type | 特性   |
|----------------------------------|------|--|
| <a href="#">Amazon Keyspaces</a> | 寬欄   | <p>以 Apache Cassandra 為基礎的高效能分散式資料庫</p> <p>使用案例包括 IoT 應用程式、事件處理和遊戲應用程式</p> |

採用專用資料庫的重要驅動因素可歸因於消除商業授權。不過，[DynamoDB](#)（包括[隨需模式](#)）、[Aurora](#)、[Amazon Neptune](#) 和 [Amazon Keyspaces](#) 等資料庫的自動擴展功能可讓您佈建平均案例的容量，而不是尖峰用量。Timestream 等專門建置的資料庫是無伺服器，可自動擴展以滿足需求，無需任何預先佈建。

AWS 如果您想要使用專門建置的開放原始碼相容關聯式資料庫，但無法或不願意對應用程式進行重大程式碼變更，會提供 [Babelfish for Aurora PostgreSQL](#)。在某些情況下，Babelfish 允許您使用現有的 SQL Server 存取碼，幾乎沒有變更。

為應用程式選擇專用關聯式資料庫時，請務必保留應用程式所需的相同（或功能同等）功能。此建議會將專用資料庫視為應用程式的主要資料存放區。特定應用程式（例如快取）會在其他建議中處理。

## 成本影響

為 .NET 工作負載採用專門建置的資料庫，雖然不太可能直接影響運算耗用量/成本，但可以直接影響 .NET 應用程式耗用的資料庫服務成本。事實上，相較於敏捷性、可擴展性、彈性和資料耐久性的額外優勢，節省成本可能是次要目標。

本指南範圍外說明為應用程式選擇專用資料庫的完整程序，以及重新建構資料策略以有效使用它們。如需詳細資訊，請參閱 AWS 教學課程目錄中的[專用資料庫](#)。

下表顯示幾個使用專用資料庫取代 SQL Server 如何改變應用程式成本的範例。請注意，這些只是粗略的預估。計算確切的生產成本需要實際工作負載的基準和最佳化。

這些是一些常用的專用資料庫預估，包括隨需運算和 100 GB SSD、中的單一執行個體資料庫 us-east-1。授權成本包括 SQL Server 授權加上軟體保證。

下表顯示商業資料庫範例的預估成本。

| 資料庫引擎                                       | 授權模式 | 執行個體類型/規格                          | AWS 運算 + 儲存成本 | 授權成本        | 每月總成本       |
|---|------|------------------------------------|---------------|-------------|-------------|
| Amazon EC2 上的 SQL Server Standard 版         | 包含授權 | r6i.2xlarge (8 個 CPU/64 GB RAM)    | 1,345.36 美元   | 0.00 美元     | 1,345.36 美元 |
| Amazon EC2 上的 SQL Server Enterprise Edition | 包含授權 | r6i.2xlarge (8 個 CPU/64 GB RAM)    | 2,834.56 美元   | 0.00 美元     | 2,834.56 美元 |
| Amazon EC2 上的 SQL Server Standard 版         | BYOL | r6i.2xlarge (8 個 CPU/64 GB RAM)    | 644.56 美元     | 456.00 美元   | 1,100.56 美元 |
| Amazon EC2 上的 SQL Server Enterprise Edition | BYOL | r6i.2xlarge (8 個 CPU/64 GB RAM)    | 644.56 美元     | 1,750.00 美元 | 2,394.56 美元 |
| Amazon RDS 上的 SQL Server Standard 版         |      | db.r6i.2xlarge (8 個 CPU/64 GB RAM) | 2,318.30 美元   | 0.00 美元     | 2,318.30 美元 |
| Amazon RDS 上的 SQL Server Enterprise Edition |      | db.r6i.2xlarge (8 個 CPU/64 GB RAM) | 3,750.56 美元   | 0.00 美元     | 3,750.56 美元 |

下表顯示專用範例的預估成本。

| 資料庫引擎                    | 執行個體類型/規格                          | AWS 運算 + 儲存成本 | 授權成本    | 每月總成本     |
|--------------------------|------------------------------------|---------------|---------|-----------|
| Amazon Aurora PostgreSQL | r6g.2xlarge (8 個 CPU/64 GB RAM)    | 855.87 美元     | 0.00 美元 | 855.87 美元 |
| DynamoDB                 | 佈建的基礎 100 WCU/400 RCU              | 72.00 美元      |         | 72.00 美元  |
| Amazon DocumentDB        | db.r6i.2xlarge (8 個 CPU/64 GB RAM) | 778.60 美元     |         | 778.60 美元 |

### ⚠ Important

資料表是根據購買前三年具有軟體保證之 SQL Server 的預估授權成本。針對 SQL Server Standard 版：4, 100 USD, 2 個核心套件, 3 年。針對 SQL Server Enterprise Edition：\$15, 700, 2 個核心套件, 3 年。

我們建議您在採用專用資料庫之前考慮成本影響。例如，更新應用程式以使用專用資料庫的成本與應用程式和來源資料庫的複雜性有關。規劃此架構切換時，請務必考量擁有總成本。這包括重構您的應用程式、提升新技術的技能，以及仔細規劃每個工作負載預期的效能和耗用。從那裡，您可以判斷投資是否值得節省成本。在大多數情況下，維護end-of-support的產品是安全和合規風險，而修復產品的成本值得投入精力和初始投資。

## 成本最佳化建議

對於存取 SQL Server 的 .NET 應用程式，有專用關聯式資料庫的替代程式庫。您可以在應用程式中實作這些程式庫，以取代類似的 SQL Server 應用程式功能。

下表重點介紹可在許多常見案例中使用的一些程式庫。

| 程式庫   | 資料庫                         | 取代                      | 架構相容性                          |
|---|-----------------------------|-------------------------|--------------------------------|
| <a href="#">Npgsql 實體架構核心提供者</a>                      | Amazon Aurora<br>PostgreSQL | 實體架構核心 SQL Server 提供者   | 現代 .NET                        |
| <a href="#">Npgsql 實體架構 6 供應商</a>                     | Amazon Aurora<br>PostgreSQL | 實體架構 6.0 SQL Server 提供者 | .NET Framework                 |
| <a href="#">Npgsql</a> ( 相容於 PostgreSQL 程式庫的 ADO.NET) | Amazon Aurora<br>PostgreSQL | ADO.NET                 | .NET Framework/<br>Modern .NET |
| <a href="#">MySQL 實體架構核心提供者</a>                       | Amazon Aurora<br>MySQL      | 實體架構核心 SQL Server 提供者   | 現代 .NET                        |
| <a href="#">Pomelo.EntityFrameworkCore.MySql</a>      | Amazon Aurora<br>MySQL      | 實體架構核心 SQL Server 提供者   | 現代 .NET                        |

使用 [Babelfish 連線至 Amazon Aurora PostgreSQL](#) 不需要任何特殊編碼即可連線。不過，所有程式碼都應該在使用前經過徹底測試。

其他專門建置的資料庫具有用於存取 .NET 相容程式庫的程式庫，可讓您存取專門建置的資料庫。範例包括：

- [使用 Amazon DynamoDB NoSQL 資料庫](#) (適用於 .NET 的 AWS SDK 文件)
- [MongoDB C# 驅動程式](#) (MongoDB 文件)
- [.NET](#) (Timestream 文件)
- [使用 Cassandra .NET Core 用戶端驅動程式以程式設計方式存取 Amazon Keyspaces](#) (Amazon Keyspaces 文件)
- [使用 .NET 連線到 Neptune 資料庫執行個體](#) (Neptune 文件)

如果您遷移至專用內建資料庫，您可以使用 中的這些工具 AWS 來協助遷移程序：

- [AWS Schema Conversion Tool \(AWS SCT\)](#) 可協助您將 SQL Server 結構描述轉換為 Amazon Aurora 和 Amazon DynamoDB。
- [AWS Database Migration Service \(AWS DMS\)](#) 可協助您將資料從 SQL Server 遷移至 Aurora 或 DynamoDB，一次或持續遷移。

- [Babelfish Compass](#) 可協助您檢查 SQL Server 資料庫的相容性，以便與 Babelfish for Aurora PostgreSQL 搭配使用。

## 其他資源

- [將 SQL Server 遷移至 Amazon Aurora PostgreSQL 的指引](#) (AWS 資料庫部落格)
- [.NET 現代化研討會](#) (AWS Workshop Studio)
- [Babelfish APP Modernization Immersion Day](#) (AWS 工作坊工作室)
- [.NET 沉浸日](#) (AWS 工作坊工作室)
- [使用 .NET \(GitHub\) 開始使用 Amazon Timestream](#) GitHub
- [上的現代 .NET 應用程式專用資料庫 AWS](#)(AWS 簡報)

## 後續步驟

檢閱完本指南後，建議您採取下列後續步驟來實作 MACO：

1. 聯絡 MACO 專家。MACO 專家可協助您回答問題並解決您的疑慮。如果您已經與 AWS 帳戶團隊合作，請聯絡團隊並請求 MACO 專家協助。如果您沒有帳戶團隊，請聯絡 [optimize-microsoft@amazon.com](mailto:optimize-microsoft@amazon.com)。
2. 套用建議。套用您在本指南中學習，以及與 MACO 專家交談的建議、最佳實務和策略。
3. 追蹤成本變更。標記您的工作負載，並使用 AWS Cost Explorer 和 等服務，AWS Budgets 進行詳細的成本追蹤、監控和控制。

# 文件歷史紀錄

下表描述了本指南的重大變更。如果您想收到有關未來更新的通知，可以訂閱 [RSS 摘要](#)。

| 變更                               | 描述  | 日期               |
|----------------------------------|---|------------------|
| <a href="#">SQL Server 更新</a>    | 我們更新了針對 <a href="#">SQL Server 工作負載最佳化 CPU</a> 一節。  | 2024 年 10 月 25 日 |
| <a href="#">SQL Server 和容器更新</a> | 我們新增了最佳化 <a href="#">SQL Server 大小</a> ，方法是使用 <a href="#">Compute Optimizer</a> 、 <a href="#">檢閱 SQL Server 工作負載 Trusted Advisor 的建議</a> ，以及使用 <a href="#">App2Container 區段來複寫 Windows 應用程式</a> 。 | 2024 年 6 月 29 日  |
| <a href="#">SQL Server 授權最佳化</a> | 我們新增了使用 <a href="#">Compute Optimizer 最佳化 SQL Server 授權</a> 一節。   | 2024 年 5 月 22 日  |
| <a href="#">初次出版</a>             | —   | 2023 年 12 月 21 日 |

# AWS 規範性指引詞彙表

以下是 AWS Prescriptive Guidance 提供的策略、指南和模式中常用的術語。若要建議項目，請使用詞彙表末尾的提供意見回饋連結。

## 數字

### 7 R

將應用程式移至雲端的七種常見遷移策略。這些策略以 Gartner 在 2011 年確定的 5 R 為基礎，包括以下內容：

- 重構/重新架構 – 充分利用雲端原生功能來移動應用程式並修改其架構，以提高敏捷性、效能和可擴展性。這通常涉及移植作業系統和資料庫。範例：將您的內部部署 Oracle 資料庫遷移至 Amazon Aurora PostgreSQL 相容版本。
- 平台轉換 (隨即重塑) – 將應用程式移至雲端，並引入一定程度的優化以利用雲端功能。範例：將您的現場部署 Oracle 資料庫遷移至 中的 Amazon Relational Database Service (Amazon RDS) for Oracle AWS 雲端。
- 重新購買 (捨棄再購買) – 切換至不同的產品，通常從傳統授權移至 SaaS 模型。範例：將您的客戶關係管理 (CRM) 系統遷移至 Salesforce.com。
- 主機轉換 (隨即轉移) – 將應用程式移至雲端，而不進行任何變更以利用雲端功能。範例：將您的現場部署 Oracle 資料庫遷移至 中 EC2 執行個體上的 Oracle AWS 雲端。
- 重新放置 (虛擬機器監視器等級隨即轉移) – 將基礎設施移至雲端，無需購買新硬體、重寫應用程式或修改現有操作。您可以將伺服器從內部部署平台遷移到相同平台的雲端服務。範例：將 Microsoft Hyper-V 應用程式遷移至 AWS。
- 保留 (重新檢視) – 將應用程式保留在來源環境中。其中可能包括需要重要重構的應用程式，且您希望將該工作延遲到以後，以及您想要保留的舊版應用程式，因為沒有業務理由來進行遷移。
- 淘汰 – 解除委任或移除來源環境中不再需要的應用程式。

## A

### ABAC

請參閱 [屬性型存取控制](#)。

## 抽象服務

請參閱 [受管服務](#)。

## ACID

請參閱 [原子性、一致性、隔離性、持久性](#)。

## 主動-主動式遷移

一種資料庫遷移方法，其中來源和目標資料庫保持同步 (透過使用雙向複寫工具或雙重寫入操作)，且兩個資料庫都在遷移期間處理來自連接應用程式的交易。此方法支援小型、受控制批次的遷移，而不需要一次性切換。它更靈活，但需要比 [主動-被動遷移](#) 更多的工作。

## 主動-被動式遷移

一種資料庫遷移方法，其中來源和目標資料庫保持同步，但只有來源資料庫處理來自連接應用程式的交易，同時將資料複寫至目標資料庫。目標資料庫在遷移期間不接受任何交易。

## 彙總函數

在一組資料列上運作的 SQL 函數，會計算群組的單一傳回值。彙總函數的範例包括 SUM 和 MAX。

## AI

請參閱 [人工智慧](#)。

## AIOps

請參閱 [人工智慧操作](#)。

## 匿名化

永久刪除資料集中個人資訊的程序。匿名化有助於保護個人隱私權。匿名資料不再被視為個人資料。

## 反模式

經常用於經常性問題的解決方案，其中解決方案具有反生產力、無效或比替代解決方案更有效。

## 應用程式控制

一種安全方法，僅允許使用核准的應用程式，以協助保護系統免受惡意軟體攻擊。

## 應用程式組合

有關組織使用的每個應用程式的詳細資訊的集合，包括建置和維護應用程式的成本及其商業價值。此資訊是 [產品組合探索和分析程序](#) 的關鍵，有助於識別要遷移、現代化和優化的應用程式並排定其優先順序。

## 人工智慧 (AI)

電腦科學領域，致力於使用運算技術來執行通常與人類相關的認知功能，例如學習、解決問題和識別模式。如需詳細資訊，請參閱[什麼是人工智慧？](#)

## 人工智慧操作 (AIOps)

使用機器學習技術解決操作問題、減少操作事件和人工干預以及提高服務品質的程序。如需有關如何在 AWS 遷移策略中使用 AIOps 的詳細資訊，請參閱[操作整合指南](#)。

## 非對稱加密

一種加密演算法，它使用一對金鑰：一個用於加密的公有金鑰和一個用於解密的私有金鑰。您可以共用公有金鑰，因為它不用於解密，但對私有金鑰存取應受到高度限制。

## 原子性、一致性、隔離性、持久性 (ACID)

一組軟體屬性，即使在出現錯誤、電源故障或其他問題的情況下，也能確保資料庫的資料有效性和操作可靠性。

## 屬性型存取控制 (ABAC)

根據使用者屬性 (例如部門、工作職責和團隊名稱) 建立精細許可的實務。如需詳細資訊，請參閱《AWS Identity and Access Management (IAM) 文件》中的[ABAC for AWS](#)。

## 授權資料來源

存放主要版本資料的位置，被視為最可靠的資訊來源。您可以將授權資料來源中的資料複製到其他位置，以處理或修改資料，例如匿名、修訂或假名化資料。

## 可用區域

中的不同位置 AWS 區域，可隔離其他可用區域中的故障，並提供相同區域中其他可用區域的低成本、低延遲網路連線。

## AWS 雲端採用架構 (AWS CAF)

的指導方針和最佳實務架構 AWS，可協助組織制定高效且有效的計劃，以成功地移至雲端。AWS CAF 將指導方針組織到六個重點領域：業務、人員、治理、平台、安全和營運。業務、人員和控管層面著重於業務技能和程序；平台、安全和操作層面著重於技術技能和程序。例如，人員層面針對處理人力資源 (HR)、人員配備功能和人員管理的利害關係人。因此，AWS CAF 為人員開發、訓練和通訊提供指引，協助組織做好成功採用雲端的準備。如需詳細資訊，請參閱[AWS CAF 網站](#)和[AWS CAF 白皮書](#)。

## AWS 工作負載資格架構 (AWS WQF)

一種工具，可評估資料庫遷移工作負載、建議遷移策略，並提供工作預估值。AWS WQF 隨附於 AWS Schema Conversion Tool (AWS SCT)。它會分析資料庫結構描述和程式碼物件、應用程式程式碼、相依性和效能特性，並提供評估報告。

## B

### 錯誤的機器人

旨在中斷或傷害個人或組織的[機器人](#)。

### BCP

請參閱[業務持續性規劃](#)。

### 行為圖

資源行為的統一互動式檢視，以及一段時間後的互動。您可以將行為圖與 Amazon Detective 搭配使用來檢查失敗的登入嘗試、可疑的 API 呼叫和類似動作。如需詳細資訊，請參閱偵測文件中的[行為圖中的資料](#)。

### 大端序系統

首先儲存最高有效位元組的系統。另請參閱 [Endianness](#)。

### 二進制分類

預測二進制結果的過程 (兩個可能的類別之一)。例如，ML 模型可能需要預測諸如「此電子郵件是否是垃圾郵件？」等問題 或「產品是書還是汽車？」

### Bloom 篩選條件

一種機率性、記憶體高效的資料結構，用於測試元素是否為集的成員。

### 藍/綠部署

一種部署策略，您可以在其中建立兩個不同但相同的環境。您可以在一個環境（藍色）中執行目前的應用程式版本，並在另一個環境（綠色）中執行新的應用程式版本。此策略可協助您快速復原，並將影響降至最低。

### 機器人

透過網際網路執行自動化任務並模擬人類活動或互動的軟體應用程式。有些機器人有用或有益，例如在網際網路上為資訊編製索引的 Web 爬蟲程式。某些其他機器人稱為惡意機器人，旨在中斷或傷害個人或組織。

## 殭屍網路

受到[惡意軟體](#)感染且受單一方控制之[機器人](#)的網路，稱為機器人繼承器或機器人運算子。殭屍網路是擴展機器人及其影響的最佳已知機制。

## 分支

程式碼儲存庫包含的區域。儲存庫中建立的第一個分支是主要分支。您可以從現有分支建立新分支，然後在新分支中開發功能或修正錯誤。您建立用來建立功能的分支通常稱為功能分支。當準備好發佈功能時，可以將功能分支合併回主要分支。如需詳細資訊，請參閱[關於分支](#) (GitHub 文件)。

## 碎片存取

在特殊情況下，以及透過核准的程序，讓使用者快速取得他們通常無權存取 AWS 帳戶 之 的存取權。如需詳細資訊，請參閱 Well-Architected 指南中的 AWS [實作打破玻璃程序](#) 指標。

## 棕地策略

環境中的現有基礎設施。對系統架構採用棕地策略時，可以根據目前系統和基礎設施的限制來設計架構。如果正在擴展現有基礎設施，則可能會混合棕地和[綠地](#)策略。

## 緩衝快取

儲存最常存取資料的記憶體區域。

## 業務能力

業務如何創造價值 (例如，銷售、客戶服務或營銷)。業務能力可驅動微服務架構和開發決策。如需詳細資訊，請參閱在 [AWS 上執行容器化微服務](#) 白皮書的[圍繞業務能力進行組織](#) 部分。

## 業務連續性規劃 (BCP)

一種解決破壞性事件 (如大規模遷移) 對營運的潛在影響並使業務能夠快速恢復營運的計畫。

# C

## CAF

請參閱[AWS 雲端採用架構](#)。

## Canary 部署

版本對最終使用者的緩慢和增量版本。當您有信心時，您可以部署新版本，並完全取代目前的版本。

## CCoE

請參閱 [Cloud Center of Excellence](#)。

## CDC

請參閱 [變更資料擷取](#)。

### 變更資料擷取 (CDC)

追蹤對資料來源 (例如資料庫表格) 的變更並記錄有關變更的中繼資料的程序。您可以將 CDC 用於各種用途，例如稽核或複寫目標系統中的變更以保持同步。

## 混沌工程

故意引入故障或破壞性事件，以測試系統的彈性。您可以使用 [AWS Fault Injection Service \(AWS FIS\)](#) 執行實驗，為您的 AWS 工作負載帶來壓力，並評估其回應。

## CI/CD

請參閱 [持續整合和持續交付](#)。

## 分類

有助於產生預測的分類程序。用於分類問題的 ML 模型可預測離散值。離散值永遠彼此不同。例如，模型可能需要評估影像中是否有汽車。

## 用戶端加密

在目標 AWS 服務接收資料之前，在本機加密資料。

## 雲端卓越中心 (CCoE)

一個多學科團隊，可推動整個組織的雲端採用工作，包括開發雲端最佳實務、調動資源、制定遷移時間表以及領導組織進行大規模轉型。如需詳細資訊，請參閱 AWS 雲端企業策略部落格上的 [CCoE 文章](#)。

## 雲端運算

通常用於遠端資料儲存和 IoT 裝置管理的雲端技術。雲端運算通常連接到 [邊緣運算](#) 技術。

## 雲端操作模型

在 IT 組織中，用於建置、成熟和最佳化一或多個雲端環境的操作模型。如需詳細資訊，請參閱 [建置您的雲端操作模型](#)。

## 採用雲端階段

組織在遷移至時通常會經歷的四個階段 AWS 雲端：

- 專案 – 執行一些與雲端相關的專案以進行概念驗證和學習用途
- 基礎 – 進行基礎投資以擴展雲端採用 (例如，建立登陸區域、定義 CCoE、建立營運模型)
- 遷移 – 遷移個別應用程式
- 重塑 – 優化產品和服務，並在雲端中創新

這些階段由 Stephen Orban 在部落格文章 [The Journey Toward Cloud-First 和 Enterprise Strategy 部落格上的採用階段](#) 中定義。AWS 雲端 如需有關它們如何與 AWS 遷移策略相關的詳細資訊，請參閱 [遷移整備指南](#)。

## CMDB

請參閱 [組態管理資料庫](#)。

## 程式碼儲存庫

透過版本控制程序來儲存及更新原始程式碼和其他資產 (例如文件、範例和指令碼) 的位置。常見的雲端儲存庫包括 GitHub 或 Bitbucket Cloud。程式碼的每個版本都稱為分支。在微服務結構中，每個儲存庫都專用於單個功能。單一 CI/CD 管道可以使用多個儲存庫。

## 冷快取

一種緩衝快取，它是空的、未填充的，或者包含過時或不相關的資料。這會影響效能，因為資料庫執行個體必須從主記憶體或磁碟讀取，這比從緩衝快取讀取更慢。

## 冷資料

很少存取且通常是歷史資料的資料。查詢這類資料時，通常可接受慢查詢。將此資料移至效能較低且成本較低的儲存層或類別，可以降低成本。

## 電腦視覺 (CV)

AI 欄位 [???](#)，使用機器學習從數位影像和影片等視覺化格式分析和擷取資訊。例如，Amazon SageMaker AI 提供 CV 的影像處理演算法。

## 組態偏離

對於工作負載，組態會從預期狀態變更。這可能會導致工作負載不合規，而且通常是漸進和無意的。

## 組態管理資料庫 (CMDB)

儲存和管理有關資料庫及其 IT 環境的資訊的儲存庫，同時包括硬體和軟體元件及其組態。您通常在遷移的產品組合探索和分析階段使用 CMDB 中的資料。

## 一致性套件

您可以組合的 AWS Config 規則和修補動作集合，以自訂您的合規和安全檢查。您可以使用 YAML 範本，將一致性套件部署為 AWS 帳戶和區域中或整個組織的單一實體。如需詳細資訊，請參閱 AWS Config 文件中的[一致性套件](#)。

## 持續整合和持續交付 (CI/CD)

自動化軟體發程序的來源、建置、測試、暫存和生產階段的程序。CI/CD 通常被描述為管道。CI/CD 可協助您將程序自動化、提升生產力、改善程式碼品質以及加快交付速度。如需詳細資訊，請參閱[持續交付的優點](#)。CD 也可表示持續部署。如需詳細資訊，請參閱[持續交付與持續部署](#)。

## CV

請參閱[電腦視覺](#)。

## D

### 靜態資料

網路中靜止的資料，例如儲存中的資料。

### 資料分類

根據重要性和敏感性來識別和分類網路資料的程序。它是所有網路安全風險管理策略的關鍵組成部分，因為它可以協助您確定適當的資料保護和保留控制。資料分類是 AWS Well-Architected Framework 中安全支柱的元件。如需詳細資訊，請參閱[資料分類](#)。

### 資料偏離

生產資料與用於訓練 ML 模型的資料之間有意義的變化，或輸入資料隨時間有意義的變更。資料偏離可以降低 ML 模型預測的整體品質、準確性和公平性。

### 傳輸中的資料

在您的網路中主動移動的資料，例如在網路資源之間移動。

### 資料網格

架構架構，提供分散式、分散式資料擁有權與集中式管理。

### 資料最小化

僅收集和處理嚴格必要資料的原則。在中實作資料最小化 AWS 雲端可以降低隱私權風險、成本和分析碳足跡。

## 資料周邊

AWS 環境中的一組預防性防護機制，可協助確保只有信任的身分才能從預期的網路存取信任的資源。如需詳細資訊，請參閱[在上建置資料周邊 AWS](#)。

## 資料預先處理

將原始資料轉換成 ML 模型可輕鬆剖析的格式。預處理資料可能意味著移除某些欄或列，並解決遺失、不一致或重複的值。

## 資料來源

在整個生命週期中追蹤資料的原始伺服器 and 歷史記錄的程序，例如資料的產生、傳輸和儲存方式。

## 資料主體

正在收集和處理其資料的個人。

## 資料倉儲

支援商業智慧的資料管理系統，例如分析。資料倉儲通常包含大量歷史資料，通常用於查詢和分析。

## 資料庫定義語言 (DDL)

用於建立或修改資料庫中資料表和物件之結構的陳述式或命令。

## 資料庫處理語言 (DML)

用於修改 (插入、更新和刪除) 資料庫中資訊的陳述式或命令。

## DDL

請參閱[資料庫定義語言](#)。

## 深度整體

結合多個深度學習模型進行預測。可以使用深度整體來獲得更準確的預測或估計預測中的不確定性。

## 深度學習

一個機器學習子領域，它使用多層人工神經網路來識別感興趣的輸入資料與目標變數之間的對應關係。

## 深度防禦

這是一種資訊安全方法，其中一系列的安全機制和控制項會在整個電腦網路中精心分層，以保護網路和其中資料的機密性、完整性和可用性。當您在 上採用此策略時 AWS，您可以在 AWS

Organizations 結構的不同層新增多個控制項，以協助保護資源。例如，defense-in-depth方法可能會結合多重要素驗證、網路分割和加密。

## 委派的管理員

在中 AWS Organizations，相容的服務可以註冊 AWS 成員帳戶，以管理組織的帳戶和管理該服務的許可。此帳戶稱為該服務的委派管理員。如需詳細資訊和相容服務清單，請參閱 AWS Organizations 文件中的[可搭配 AWS Organizations運作的服務](#)。

## 部署

在目標環境中提供應用程式、新功能或程式碼修正的程序。部署涉及在程式碼庫中實作變更，然後在應用程式環境中建置和執行該程式碼庫。

## 開發環境

請參閱[環境](#)。

## 偵測性控制

一種安全控制，用於在事件發生後偵測、記錄和提醒。這些控制是第二道防線，提醒您注意繞過現有預防性控制的安全事件。如需詳細資訊，請參閱在 AWS 上實作安全控制中的[偵測性控制](#)。

## 開發值串流映射 (DVSM)

一種程序，用於識別對軟體開發生命週期中的速度和品質造成負面影響的限制並排定優先順序。DVSM 延伸了原本專為精簡製造實務設計的價值串流映射程序。它著重於透過軟體開發程序建立和移動價值所需的步驟和團隊。

## 數位分身

真實世界系統的虛擬呈現，例如建築物、工廠、工業設備或生產線。數位分身支援預測性維護、遠端監控和生產最佳化。

## 維度資料表

在[星星結構描述](#)中，較小的資料表包含有關事實資料表中量化資料的資料屬性。維度資料表屬性通常是文字欄位或離散數字，其行為類似於文字。這些屬性通常用於查詢限制、篩選和結果集標記。

## 災難

防止工作負載或系統在其主要部署位置中實現其業務目標的事件。這些事件可能是自然災難、技術故障或人為動作的結果，例如意外設定錯誤或惡意軟體攻擊。

## 災難復原 (DR)

您用來將[災難](#)造成的停機時間和資料遺失降至最低的策略和程序。如需詳細資訊，請參閱 AWS Well-Architected Framework 中的[上工作負載災難復原 AWS：雲端中的復原](#)。

## DML

請參閱[資料庫處理語言](#)。

### 領域驅動的設計

一種開發複雜軟體系統的方法，它會將其元件與每個元件所服務的不斷發展的領域或核心業務目標相關聯。Eric Evans 在其著作 *Domain-Driven Design: Tackling Complexity in the Heart of Software* (Boston: Addison-Wesley Professional, 2003) 中介紹了這一概念。如需有關如何將領域驅動的設計與 strangler fig 模式搭配使用的資訊，請參閱[使用容器和 Amazon API Gateway 逐步現代化舊版 Microsoft ASP.NET \(ASMX\) Web 服務](#)。

## DR

請參閱[災難復原](#)。

### 偏離偵測

追蹤與基準組態的偏差。例如，您可以使用 AWS CloudFormation 來偵測系統資源中的偏離，也可以使用 AWS Control Tower 來[偵測登陸區域中可能影響控管要求合規性的變更](#)。<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/using-cfn-stack-drift.html>

## DVSM

請參閱[開發值串流映射](#)。

## E

### EDA

請參閱[探索性資料分析](#)。

### EDI

請參閱[電子資料交換](#)。

### 邊緣運算

提升 IoT 網路邊緣智慧型裝置運算能力的技術。與[雲端運算](#)相比，邊緣運算可以減少通訊延遲並改善回應時間。

### 電子資料交換 (EDI)

在組織之間自動交換商業文件。如需詳細資訊，請參閱[什麼是電子資料交換](#)。

## 加密

一種運算程序，可將人類可讀取的純文字資料轉換為加密文字。

### 加密金鑰

由加密演算法產生的隨機位元的加密字串。金鑰長度可能有所不同，每個金鑰的設計都是不可預測且唯一的。

### 端序

位元組在電腦記憶體中的儲存順序。大端序系統首先儲存最高有效位元組。小端序系統首先儲存最低有效位元組。

### 端點

請參閱 [服務端點](#)。

### 端點服務

您可以在虛擬私有雲端 (VPC) 中託管以與其他使用者共用的服務。您可以使用 [建立端點服務](#)，AWS PrivateLink 並將許可授予其他 AWS 帳戶 或 AWS Identity and Access Management (IAM) 委託人。這些帳戶或主體可以透過建立介面 VPC 端點私下連接至您的端點服務。如需詳細資訊，請參閱 Amazon Virtual Private Cloud (Amazon VPC) 文件中的 [建立端點服務](#)。

### 企業資源規劃 (ERP)

一種系統，可自動化和管理企業的關鍵業務流程（例如會計、[MES](#) 和專案管理）。

### 信封加密

使用另一個加密金鑰對某個加密金鑰進行加密的程序。如需詳細資訊，請參閱 AWS Key Management Service (AWS KMS) 文件中的 [信封加密](#)。

### 環境

執行中應用程式的執行個體。以下是雲端運算中常見的環境類型：

- 開發環境 – 執行中應用程式的執行個體，只有負責維護應用程式的核心團隊才能使用。開發環境用來測試變更，然後再將開發環境提升到較高的環境。此類型的環境有時稱為測試環境。
- 較低的環境 – 應用程式的所有開發環境，例如用於初始建置和測試的開發環境。
- 生產環境 – 最終使用者可以存取的執行中應用程式的執行個體。在 CI/CD 管道中，生產環境是最後一個部署環境。
- 較高的環境 – 核心開發團隊以外的使用者可存取的所有環境。這可能包括生產環境、生產前環境以及用於使用者接受度測試的環境。

## epic

在敏捷方法中，有助於組織工作並排定工作優先順序的功能類別。epic 提供要求和實作任務的高層級描述。例如，AWS CAF 安全概念包括身分和存取管理、偵測控制、基礎設施安全、資料保護和事件回應。如需有關 AWS 遷移策略中的 Epic 的詳細資訊，請參閱[計畫實作指南](#)。

## ERP

請參閱[企業資源規劃](#)。

## 探索性資料分析 (EDA)

分析資料集以了解其主要特性的過程。您收集或彙總資料，然後執行初步調查以尋找模式、偵測異常並檢查假設。透過計算摘要統計並建立資料可視化來執行 EDA。

## F

### 事實資料表

[星狀結構描述](#)中的中央資料表。它存放有關業務操作的量化資料。一般而言，事實資料表包含兩種類型的資料欄：包含度量的資料，以及包含維度資料表外部索引鍵的資料欄。

### 快速失敗

一種使用頻繁和增量測試來縮短開發生命週期的理念。這是敏捷方法的關鍵部分。

### 故障隔離界限

在中 AWS 雲端，像是可用區域 AWS 區域、控制平面或資料平面等邊界會限制故障的影響，並有助於改善工作負載的彈性。如需詳細資訊，請參閱[AWS 故障隔離界限](#)。

### 功能分支

請參閱[分支](#)。

### 特徵

用來進行預測的輸入資料。例如，在製造環境中，特徵可能是定期從製造生產線擷取的影像。

### 功能重要性

特徵對於模型的預測有多重要。這通常表示為可以透過各種技術來計算的數值得分，例如 Shapley Additive Explanations (SHAP) 和積分梯度。如需詳細資訊，請參閱[機器學習模型可解譯性 AWS](#)。

## 特徵轉換

優化 ML 程序的資料，包括使用其他來源豐富資料、調整值、或從單一資料欄位擷取多組資訊。這可讓 ML 模型從資料中受益。例如，如果將「2021-05-27 00:15:37」日期劃分為「2021」、「五月」、「週四」和「15」，則可以協助學習演算法學習與不同資料元件相關聯的細微模式。

### 少量擷取提示

在要求 [LLM](#) 執行類似的任務之前，提供少量示範任務和所需輸出的範例。此技術是內容內學習的應用程式，其中模型會從內嵌在提示中的範例 (快照) 中學習。對於需要特定格式、推理或網域知識的任務，少量的提示非常有效。另請參閱[零鏡頭提示](#)。

## FGAC

請參閱[精細存取控制](#)。

### 精細存取控制 (FGAC)

使用多個條件來允許或拒絕存取請求。

### 閃切遷移

一種資料庫遷移方法，透過[變更資料擷取](#)使用連續資料複寫，以盡可能在最短的時間內遷移資料，而不是使用分階段方法。目標是將停機時間降至最低。

## FM

請參閱[基礎模型](#)。

### 基礎模型 (FM)

大型深度學習神經網路，已針對廣義和未標記資料的大量資料集進行訓練。FMs 能夠執行各種一般任務，例如了解語言、產生文字和影像，以及以自然語言交談。如需詳細資訊，請參閱[什麼是基礎模型](#)。

## G

### 生成式 AI

已針對大量資料進行訓練的 [AI](#) 模型子集，可使用簡單的文字提示建立新的內容和成品，例如影像、影片、文字和音訊。如需詳細資訊，請參閱[什麼是生成式 AI](#)。

### 地理封鎖

請參閱[地理限制](#)。

## 地理限制 (地理封鎖)

Amazon CloudFront 中的選項，可防止特定國家/地區的使用者存取內容分發。您可以使用允許清單或封鎖清單來指定核准和禁止的國家/地區。如需詳細資訊，請參閱 CloudFront 文件中的[限制內容的地理分佈](#)。

## Gitflow 工作流程

這是一種方法，其中較低和較高環境在原始碼儲存庫中使用不同分支。Gitflow 工作流程會被視為舊版，而以[幹線為基礎的工作流程](#)是現代、偏好的方法。

## 黃金影像

系統或軟體的快照，做為部署該系統或軟體新執行個體的範本。例如，在製造中，黃金映像可用於在多個裝置上佈建軟體，並有助於提高裝置製造操作的速度、可擴展性和生產力。

## 綠地策略

新環境中缺乏現有基礎設施。對系統架構採用綠地策略時，可以選擇所有新技術，而不會限制與現有基礎設施的相容性，也稱為[棕地](#)。如果正在擴展現有基礎設施，則可能會混合棕地和綠地策略。

## 防護機制

有助於跨組織單位 (OU) 來管控資源、政策和合規的高層級規則。預防性防護機制會強制執行政策，以確保符合合規標準。透過使用服務控制政策和 IAM 許可界限來將其實作。偵測性防護機制可偵測政策違規和合規問題，並產生提醒以便修正。它們是透過使用 AWS Config AWS Security Hub、Amazon GuardDuty、Amazon Inspector AWS Trusted Advisor和自訂 AWS Lambda 檢查來實作。

# H

## HA

請參閱[高可用性](#)。

## 異質資料庫遷移

將來源資料庫遷移至使用不同資料庫引擎的目標資料庫 (例如，Oracle 至 Amazon Aurora)。異質遷移通常是重新架構工作的一部分，而轉換結構描述可能是一項複雜任務。[AWS 提供有助於結構描述轉換的 AWS SCT](#)。

## 高可用性 (HA)

在遇到挑戰或災難時，工作負載能夠在不介入的情況下持續運作。HA 系統的設計目的是自動容錯移轉、持續提供高品質的效能，並處理不同的負載和故障，並將效能影響降至最低。

## 歷史現代化

一種方法，用於現代化和升級操作技術 (OT) 系統，以更好地滿足製造業的需求。歷史資料是一種資料庫，用於從工廠中的各種來源收集和存放資料。

### 保留資料

從用於訓練機器學習模型的資料集中保留的部分歷史標記資料。您可以使用保留資料，透過比較模型預測與保留資料來評估模型效能。

### 異質資料庫遷移

將您的來源資料庫遷移至共用相同資料庫引擎的目標資料庫 (例如，Microsoft SQL Server 至 Amazon RDS for SQL Server)。同質遷移通常是主機轉換或平台轉換工作的一部分。您可以使用原生資料庫公用程式來遷移結構描述。

### 熱資料

經常存取的資料，例如即時資料或最近的轉譯資料。此資料通常需要高效能儲存層或類別，才能提供快速的查詢回應。

### 修補程序

緊急修正生產環境中的關鍵問題。由於其緊迫性，通常會在典型 DevOps 發行工作流程之外執行修補程式。

### 超級護理期間

在切換後，遷移團隊在雲端管理和監控遷移的應用程式以解決任何問題的時段。通常，此期間的長度為 1-4 天。在超級護理期間結束時，遷移團隊通常會將應用程式的責任轉移給雲端營運團隊。

## I

### IaC

將[基礎設施視為程式碼](#)。

### 身分型政策

連接至一或多個 IAM 主體的政策，可定義其在 AWS 雲端環境中的許可。

### 閒置應用程式

90 天期間 CPU 和記憶體平均使用率在 5% 至 20% 之間的應用程式。在遷移專案中，通常會淘汰這些應用程式或將其保留在內部部署。

## IloT

請參閱[工業物聯網](#)。

### 不可變的基礎設施

為生產工作負載部署新基礎設施的模型，而不是更新、修補或修改現有的基礎設施。不可變基礎設施本質上比[可變基礎設施](#)更一致、可靠且可預測。如需詳細資訊，請參閱 AWS Well-Architected Framework [中的使用不可變基礎設施部署](#)最佳實務。

### 傳入 (輸入) VPC

在 AWS 多帳戶架構中，接受、檢查和路由來自應用程式外部之網路連線的 VPC。[AWS 安全參考架構](#)建議您使用傳入、傳出和檢查 VPC 來設定網路帳戶，以保護應用程式與更廣泛的網際網路之間的雙向介面。

### 增量遷移

一種切換策略，您可以在其中將應用程式分成小部分遷移，而不是執行單一、完整的切換。例如，您最初可能只將一些微服務或使用者移至新系統。確認所有項目都正常運作之後，您可以逐步移動其他微服務或使用者，直到可以解除委任舊式系統。此策略可降低與大型遷移關聯的風險。

### 工業 4.0

2016 年 [Klaus Schwab](#) 推出的術語，透過連線能力、即時資料、自動化、分析和 AI/ML 的進展，指製造程序的現代化。

### 基礎設施

應用程式環境中包含的所有資源和資產。

### 基礎設施即程式碼 (IaC)

透過一組組態檔案來佈建和管理應用程式基礎設施的程序。IaC 旨在協助您集中管理基礎設施，標準化資源並快速擴展，以便新環境可重複、可靠且一致。

### 工業物聯網 (IIoT)

在製造業、能源、汽車、醫療保健、生命科學和農業等產業領域使用網際網路連線的感測器和裝置。如需詳細資訊，請參閱[建立工業物聯網 \(IIoT\) 數位轉型策略](#)。

### 檢查 VPC

在 AWS 多帳戶架構中，集中式 VPC 可管理 VPCs 之間（在相同或不同的 AWS 區域）、網際網路和內部部署網路之間的網路流量檢查。[AWS 安全參考架構](#)建議您使用傳入、傳出和檢查 VPC 來設定網路帳戶，以保護應用程式與更廣泛的網際網路之間的雙向介面。

## 物聯網 (IoT)

具有內嵌式感測器或處理器的相連實體物體網路，其透過網際網路或本地通訊網路與其他裝置和系統進行通訊。如需詳細資訊，請參閱[什麼是 IoT？](#)

### 可解釋性

機器學習模型的一個特徵，描述了人類能夠理解模型的預測如何依賴於其輸入的程度。如需詳細資訊，請參閱[的機器學習模型可解釋性 AWS](#)。

## IoT

請參閱[物聯網](#)。

## IT 資訊庫 (ITIL)

一組用於交付 IT 服務並使這些服務與業務需求保持一致的最佳實務。ITIL 為 ITSM 提供了基礎。

## IT 服務管理 (ITSM)

與組織的設計、實作、管理和支援 IT 服務關聯的活動。如需有關將雲端操作與 ITSM 工具整合的資訊，請參閱[操作整合指南](#)。

## ITIL

請參閱[IT 資訊庫](#)。

## ITSM

請參閱[IT 服務管理](#)。

## L

## 標籤型存取控制 (LBAC)

強制存取控制 (MAC) 的實作，其中使用者和資料本身都會獲得明確指派的安全標籤值。使用者安全標籤和資料安全標籤之間的交集會決定使用者可以看到哪些資料列和資料欄。

## 登陸區域

登陸區域是架構良好的多帳戶 AWS 環境，可擴展且安全。這是一個起點，您的組織可以從此起點快速啟動和部署工作負載與應用程式，並對其安全和基礎設施環境充滿信心。如需有關登陸區域的詳細資訊，請參閱[設定安全且可擴展的多帳戶 AWS 環境](#)。

## 大型語言模型 (LLM)

預先訓練大量資料的深度學習 [AI](#) 模型。LLM 可以執行多個任務，例如回答問題、摘要文件、將文字翻譯成其他語言，以及完成句子。如需詳細資訊，請參閱[什麼是 LLMs](#)。

### 大型遷移

遷移 300 部或更多伺服器。

### LBAC

請參閱[標籤型存取控制](#)。

### 最低權限

授予執行任務所需之最低許可的安全最佳實務。如需詳細資訊，請參閱 IAM 文件中的[套用最低權限許可](#)。

### 隨即轉移

請參閱 [7 個 R](#)。

### 小端序系統

首先儲存最低有效位元組的系統。另請參閱 [Endianness](#)。

## LLM

請參閱[大型語言模型](#)。

### 較低的環境

請參閱 [環境](#)。

## M

### 機器學習 (ML)

一種使用演算法和技術進行模式識別和學習的人工智慧。機器學習會進行分析並從記錄的資料 (例如物聯網 (IoT) 資料) 中學習，以根據模式產生統計模型。如需詳細資訊，請參閱[機器學習](#)。

### 主要分支

請參閱[分支](#)。

## 惡意軟體

旨在危及電腦安全或隱私權的軟體。惡意軟體可能會中斷電腦系統、洩露敏感資訊，或取得未經授權的存取。惡意軟體的範例包括病毒、蠕蟲、勒索軟體、特洛伊木馬、間諜軟體和鍵盤記錄器。

## 受管服務

AWS 服務會 AWS 操作基礎設施層、作業系統和平台，而您會存取端點來存放和擷取資料。Amazon Simple Storage Service (Amazon S3) 和 Amazon DynamoDB 是受管服務的範例。這些也稱為抽象服務。

## 製造執行系統 (MES)

一種軟體系統，用於追蹤、監控、記錄和控制生產程序，將原物料轉換為現場成品。

## MAP

請參閱[遷移加速計劃](#)。

## 機制

建立工具、推動工具採用，然後檢查結果以進行調整的完整程序。機制是在操作時強化和改善自身的循環。如需詳細資訊，請參閱 AWS Well-Architected Framework 中的[建置機制](#)。

## 成員帳戶

除了屬於組織一部分的管理帳戶 AWS 帳戶 之外的所有 AWS Organizations。一個帳戶一次只能是一個組織的成員。

## 製造執行系統

請參閱[製造執行系統](#)。

## 訊息佇列遙測傳輸 (MQTT)

根據[發佈/訂閱](#)模式的輕量型machine-to-machine(M2M) 通訊協定，適用於資源受限的 [IoT](#) 裝置。

## 微服務

一種小型的獨立服務，它可透過定義明確的 API 進行通訊，通常由小型獨立團隊擁有。例如，保險系統可能包含對應至業務能力 (例如銷售或行銷) 或子領域 (例如購買、索賠或分析) 的微服務。微服務的優點包括靈活性、彈性擴展、輕鬆部署、可重複使用的程式碼和適應力。如需詳細資訊，請參閱[使用無 AWS 伺服器服務整合微服務](#)。

## 微服務架構

一種使用獨立元件來建置應用程式的方法，這些元件會以微服務形式執行每個應用程式程序。這些微服務會使用輕量型 API，透過明確定義的介面進行通訊。此架構中的每個微服務都可以進行

更新、部署和擴展，以滿足應用程式特定功能的需求。如需詳細資訊，請參閱[在上實作微服務 AWS](#)。

## Migration Acceleration Program (MAP)

一種 AWS 計畫，提供諮詢支援、訓練和服務，協助組織建立強大的營運基礎，以移至雲端，並協助抵銷遷移的初始成本。MAP 包括用於有條不紊地執行舊式遷移的遷移方法以及一組用於自動化和加速常見遷移案例的工具。

### 大規模遷移

將大部分應用程式組合依波次移至雲端的程序，在每個波次中，都會以更快的速度移動更多應用程式。此階段使用從早期階段學到的最佳實務和經驗教訓來實作團隊、工具和流程的遷移工廠，以透過自動化和敏捷交付簡化工作負載的遷移。這是[AWS 遷移策略](#)的第三階段。

### 遷移工廠

可透過自動化、敏捷的方法簡化工作負載遷移的跨職能團隊。遷移工廠團隊通常包括營運、業務分析師和擁有者、遷移工程師、開發人員以及從事 Sprint 工作的 DevOps 專業人員。20% 至 50% 之間的企業應用程式組合包含可透過工廠方法優化的重複模式。如需詳細資訊，請參閱此內容集中的[遷移工廠的討論](#)和[雲端遷移工廠指南](#)。

### 遷移中繼資料

有關完成遷移所需的應用程式和伺服器的資訊。每種遷移模式都需要一組不同的遷移中繼資料。遷移中繼資料的範例包括目標子網路、安全群組和 AWS 帳戶。

### 遷移模式

可重複的遷移任務，詳細描述遷移策略、遷移目的地以及所使用的遷移應用程式或服務。範例：使用 AWS Application Migration Service 重新託管遷移至 Amazon EC2。

### 遷移組合評定 (MPA)

線上工具，提供驗證商業案例以遷移至的資訊 AWS 雲端。MPA 提供詳細的組合評定 (伺服器適當規模、定價、總體擁有成本比較、遷移成本分析) 以及遷移規劃 (應用程式資料分析和資料收集、應用程式分組、遷移優先順序，以及波次規劃)。[MPA 工具](#) (需要登入) 可供所有 AWS 顧問和 APN 合作夥伴顧問免費使用。

### 遷移準備程度評定 (MRA)

使用 AWS CAF 取得組織雲端整備狀態的洞見、識別優缺點，以及建立行動計劃以消除已識別差距的程序。如需詳細資訊，請參閱[遷移準備程度指南](#)。MRA 是[AWS 遷移策略](#)的第一階段。

## 遷移策略

用來將工作負載遷移至的方法 AWS 雲端。如需詳細資訊，請參閱此詞彙表中的 [7 個 Rs](#) 項目，並請參閱[動員您的組織以加速大規模遷移](#)。

## 機器學習 (ML)

請參閱[機器學習](#)。

## 現代化

將過時的 (舊版或單一) 應用程式及其基礎架構轉換為雲端中靈活、富有彈性且高度可用的系統，以降低成本、提高效率並充分利用創新。如需詳細資訊，請參閱 [《》中的現代化應用程式的策略 AWS 雲端](#)。

## 現代化準備程度評定

這項評估可協助判斷組織應用程式的現代化準備程度；識別優點、風險和相依性；並確定組織能夠在多大程度上支援這些應用程式的未來狀態。評定的結果就是目標架構的藍圖、詳細說明現代化程序的開發階段和里程碑的路線圖、以及解決已發現的差距之行動計畫。如需詳細資訊，請參閱 [《》中的評估應用程式的現代化準備 AWS 雲端](#) 程度。

## 單一應用程式 (單一)

透過緊密結合的程序作為單一服務執行的應用程式。單一應用程式有幾個缺點。如果一個應用程式功能遇到需求激增，則必須擴展整個架構。當程式碼庫增長時，新增或改進單一應用程式的功能也會變得更加複雜。若要解決這些問題，可以使用微服務架構。如需詳細資訊，請參閱[將單一體系分解為微服務](#)。

## MPA

請參閱[遷移產品組合評估](#)。

## MQTT

請參閱[訊息佇列遙測傳輸](#)。

## 多類別分類

一個有助於產生多類別預測的過程 (預測兩個以上的結果之一)。例如，機器學習模型可能會詢問「此產品是書籍、汽車還是電話？」或者「這個客戶對哪種產品類別最感興趣？」

## 可變基礎設施

更新和修改生產工作負載現有基礎設施的模型。為了提高一致性、可靠性和可預測性，AWS Well-Architected Framework 建議使用[不可變基礎設施](#)做為最佳實務。

## O

### OAC

請參閱[原始存取控制](#)。

### OAI

請參閱[原始存取身分](#)。

### OCM

請參閱[組織變更管理](#)。

### 離線遷移

一種遷移方法，可在遷移過程中刪除來源工作負載。此方法涉及延長停機時間，通常用於小型非關鍵工作負載。

### OI

請參閱[操作整合](#)。

### OLA

請參閱[操作層級協議](#)。

### 線上遷移

一種遷移方法，無需離線即可將來源工作負載複製到目標系統。連接至工作負載的應用程式可在遷移期間繼續運作。此方法涉及零至最短停機時間，通常用於關鍵的生產工作負載。

### OPC-UA

請參閱[開放程序通訊 - 統一架構](#)。

### 開放程序通訊 - 統一架構 (OPC-UA)

用於工業自動化的machine-to-machine(M2M) 通訊協定。OPC-UA 提供資料加密、身分驗證和授權機制的互通性標準。

### 操作水準協議 (OLA)

一份協議，闡明 IT 職能群組承諾向彼此提供的內容，以支援服務水準協議 (SLA)。

### 操作整備審查 (ORR)

問題和相關最佳實務的檢查清單，可協助您了解、評估、預防或減少事件和可能失敗的範圍。如需詳細資訊，請參閱 AWS Well-Architected Framework 中的[操作準備度審查 \(ORR\)](#)。

## 操作技術 (OT)

使用實體環境控制工業操作、設備和基礎設施的硬體和軟體系統。在製造業中，整合 OT 和資訊技術 (IT) 系統是[工業 4.0](#) 轉型的關鍵重點。

## 操作整合 (OI)

在雲端中將操作現代化的程序，其中包括準備程度規劃、自動化和整合。如需詳細資訊，請參閱[操作整合指南](#)。

## 組織追蹤

由建立的線索 AWS CloudTrail 會記錄 AWS 帳戶 組織中所有 的所有事件 AWS Organizations。在屬於組織的每個 AWS 帳戶 中建立此追蹤，它會跟蹤每個帳戶中的活動。如需詳細資訊，請參閱 CloudTrail 文件中的[建立組織追蹤](#)。

## 組織變更管理 (OCM)

用於從人員、文化和領導力層面管理重大、顛覆性業務轉型的架構。OCM 透過加速變更採用、解決過渡問題，以及推動文化和組織變更，協助組織為新系統和策略做好準備，並轉移至新系統和策略。在 AWS 遷移策略中，此架構稱為人員加速，因為雲端採用專案所需的變更速度。如需詳細資訊，請參閱[OCM 指南](#)。

## 原始存取控制 (OAC)

CloudFront 中的增強型選項，用於限制存取以保護 Amazon Simple Storage Service (Amazon S3) 內容。OAC 支援所有 S3 儲存貯體中的所有伺服器端加密 AWS KMS (SSE-KMS) AWS 區域，以及對 S3 儲存貯體的動態PUT和DELETE請求。

## 原始存取身分 (OAI)

CloudFront 中的一個選項，用於限制存取以保護 Amazon S3 內容。當您使用 OAI 時，CloudFront 會建立一個可供 Amazon S3 進行驗證的主體。經驗證的主體只能透過特定 CloudFront 分發來存取 S3 儲存貯體中的內容。另請參閱[OAC](#)，它可提供更精細且增強的存取控制。

## ORR

請參閱[操作整備審核](#)。

## OT

請參閱[操作技術](#)。

## 傳出 (輸出) VPC

在 AWS 多帳戶架構中，處理從應用程式內啟動之網路連線的 VPC。[AWS 安全參考架構](#)建議您使用傳入、傳出和檢查 VPC 來設定網路帳戶，以保護應用程式與更廣泛的網際網路之間的雙向介面。

## P

### 許可界限

附接至 IAM 主體的 IAM 管理政策，可設定使用者或角色擁有的最大許可。如需詳細資訊，請參閱 IAM 文件中的[許可界限](#)。

### 個人身分識別資訊 (PII)

直接檢視或與其他相關資料配對時，可用來合理推斷個人身分的資訊。PII 的範例包括名稱、地址和聯絡資訊。

### PII

請參閱[個人身分識別資訊](#)。

### 手冊

一組預先定義的步驟，可擷取與遷移關聯的工作，例如在雲端中提供核心操作功能。手冊可以採用指令碼、自動化執行手冊或操作現代化環境所需的程序或步驟摘要的形式。

### PLC

請參閱[可程式設計邏輯控制器](#)。

### PLM

請參閱[產品生命週期管理](#)。

### 政策

可定義許可的物件（請參閱[身分型政策](#)）、指定存取條件（請參閱[資源型政策](#)），或定義組織中所有帳戶的最大許可 AWS Organizations（請參閱[服務控制政策](#)）。

### 混合持久性

根據資料存取模式和其他需求，獨立選擇微服務的資料儲存技術。如果您的微服務具有相同的資料儲存技術，則其可能會遇到實作挑戰或效能不佳。如果微服務使用最適合其需求的資料儲存，則

可以更輕鬆地實作並達到更好的效能和可擴展性。如需詳細資訊，請參閱[在微服務中啟用資料持久性](#)。

## 組合評定

探索、分析應用程式組合並排定其優先順序以規劃遷移的程序。如需詳細資訊，請參閱[評估遷移準備程度](#)。

## 述詞

傳回 true 或的查詢條件 false，通常位於 WHERE 子句中。

## 述詞下推

一種資料庫查詢最佳化技術，可在傳輸前篩選查詢中的資料。這可減少必須從關聯式資料庫擷取和處理的資料量，並改善查詢效能。

## 預防性控制

旨在防止事件發生的安全控制。這些控制是第一道防線，可協助防止對網路的未經授權存取或不必要變更。如需詳細資訊，請參閱在 AWS 上實作安全控制中的[預防性控制](#)。

## 委託人

中可執行動作和存取資源 AWS 的實體。此實體通常是 AWS 帳戶、IAM 角色或使用者的根使用者。如需詳細資訊，請參閱 IAM 文件中[角色術語和概念](#)中的主體。

## 依設計的隱私權

透過整個開發程序將隱私權納入考量的系統工程方法。

## 私有託管區域

一種容器，它包含有關您希望 Amazon Route 53 如何回應一個或多個 VPC 內的域及其子域之 DNS 查詢的資訊。如需詳細資訊，請參閱 Route 53 文件中的[使用私有託管區域](#)。

## 主動控制

旨在防止部署不合規資源的[安全控制](#)。這些控制項會在佈建資源之前對其進行掃描。如果資源不符合控制項，則不會佈建。如需詳細資訊，請參閱 AWS Control Tower 文件中的[控制項參考指南](#)，並參閱實作安全[控制項中的主動](#)控制項。 AWS

## 產品生命週期管理 (PLM)

管理產品整個生命週期的資料和程序，從設計、開發和啟動，到成長和成熟，再到拒絕和移除。

## 生產環境

請參閱[環境](#)。

## 可程式設計邏輯控制器 (PLC)

在製造中，高度可靠、可調整的電腦，可監控機器並自動化製造程序。

### 提示鏈結

使用一個 [LLM](#) 提示的輸出做為下一個提示的輸入，以產生更好的回應。此技術用於將複雜任務分解為子任務，或反覆精簡或展開初步回應。它有助於提高模型回應的準確性和相關性，並允許更精細、個人化的結果。

### 擬匿名化

將資料集中的個人識別符取代為預留位置值的程序。假名化有助於保護個人隱私權。假名化資料仍被視為個人資料。

### 發佈/訂閱 (pub/sub)

一種模式，可啟用微服務之間的非同步通訊，以提高可擴展性和回應能力。例如，在微服務型 [MES](#) 中，微服務可以將事件訊息發佈到其他微服務可訂閱的頻道。系統可以新增新的微服務，而無需變更發佈服務。

## Q

### 查詢計劃

一系列步驟，如指示，用於存取 SQL 關聯式資料庫系統中的資料。

### 查詢計劃迴歸

在資料庫服務優化工具選擇的計畫比對資料庫環境進行指定的變更之前的計畫不太理想時。這可能因為對統計資料、限制條件、環境設定、查詢參數繫結的變更以及資料庫引擎的更新所導致。

## R

### RACI 矩陣

請參閱 [負責、負責、諮詢、告知 \(RACI\)](#)。

### RAG

請參閱 [擷取增強產生](#)。

## 勒索軟體

一種惡意軟體，旨在阻止對計算機系統或資料的存取，直到付款為止。

## RASCI 矩陣

請參閱[負責、負責、諮詢、告知 \(RACI\)](#)。

## RCAC

請參閱[資料列和資料欄存取控制](#)。

## 僅供讀取複本

用於唯讀用途的資料庫複本。您可以將查詢路由至僅供讀取複本以減少主資料庫的負載。

## 重新架構師

請參閱[7 個 R](#)。

## 復原點目標 (RPO)

自上次資料復原點以來可接受的時間上限。這會決定最後一個復原點與服務中斷之間可接受的資料遺失。

## 復原時間目標 (RTO)

服務中斷與服務還原之間的可接受延遲上限。

## 重構

請參閱[7 個 R](#)。

## 區域

地理區域中的 AWS 資源集合。每個 AWS 區域 都獨立於其他，以提供容錯能力、穩定性和彈性。如需詳細資訊，請參閱[指定 AWS 區域 您的帳戶可以使用哪些](#)。

## 迴歸

預測數值的 ML 技術。例如，為了解決「這房子會賣什麼價格？」的問題 ML 模型可以使用線性迴歸模型，根據已知的房屋事實 (例如，平方英尺) 來預測房屋的銷售價格。

## 重新託管

請參閱[7 個 R](#)。

## 版本

在部署程序中，它是將變更提升至生產環境的動作。

## 重新定位

請參閱 [7 個 R](#)。

## Replatform

請參閱 [7 個 R](#)。

## 回購

請參閱 [7 個 R](#)。

## 彈性

應用程式抵禦中斷或從中斷中復原的能力。[在中規劃彈性時，高可用性和災難復原](#)是常見的考量 AWS 雲端。如需詳細資訊，請參閱[AWS 雲端 彈性](#)。

## 資源型政策

附接至資源的政策，例如 Amazon S3 儲存貯體、端點或加密金鑰。這種類型的政策會指定允許存取哪些主體、支援的動作以及必須滿足的任何其他條件。

## 負責者、當責者、事先諮詢者和事後告知者 (RACI) 矩陣

矩陣，定義所有涉及遷移活動和雲端操作之各方的角色和責任。矩陣名稱衍生自矩陣中定義的責任類型：負責人 (R)、責任 (A)、已諮詢 (C) 和知情 (I)。支援 (S) 類型為選用。如果您包含支援，則矩陣稱為 RASCI 矩陣，如果您排除它，則稱為 RACI 矩陣。

## 回應性控制

一種安全控制，旨在驅動不良事件或偏離安全基準的補救措施。如需詳細資訊，請參閱在 AWS 上實作安全控制中的[回應性控制](#)。

## 保留

請參閱 [7 個 R](#)。

## 淘汰

請參閱 [7 Rs](#)。

## 檢索增強生成 (RAG)

[一種生成式 AI 技術](#)，其中 [LLM](#) 會在產生回應之前參考訓練資料來源以外的授權資料來源。例如，RAG 模型可能會對組織的知識庫或自訂資料執行語意搜尋。如需詳細資訊，請參閱[什麼是 RAG](#)。

## 輪換

定期更新[秘密](#)的程序，讓攻擊者更難存取登入資料。

## 資料列和資料欄存取控制 (RCAC)

使用已定義存取規則的基本、彈性 SQL 表達式。RCAC 包含資料列許可和資料欄遮罩。

## RPO

請參閱[復原點目標](#)。

## RTO

請參閱[復原時間目標](#)。

## 執行手冊

執行特定任務所需的一組手動或自動程序。這些通常是為了簡化重複性操作或錯誤率較高的程序而建置。

# S

## SAML 2.0

許多身分提供者 (IdP) 使用的開放標準。此功能會啟用聯合單一登入 (SSO)，讓使用者可以登入 AWS Management Console 或呼叫 AWS API 操作，而不必為您組織中的每個人在 IAM 中建立使用者。如需有關以 SAML 2.0 為基礎的聯合詳細資訊，請參閱 IAM 文件中的[關於以 SAML 2.0 為基礎的聯合](#)。

## SCADA

請參閱[監督控制和資料擷取](#)。

## SCP

請參閱[服務控制政策](#)。

## 秘密

您以加密形式存放的 AWS Secrets Manager 機密或限制資訊，例如密碼或使用者登入資料。它由秘密值及其中繼資料組成。秘密值可以是二進位、單一字串或多個字串。如需詳細資訊，請參閱 [Secrets Manager 文件中的 Secrets Manager 秘密中的什麼內容？](#)。

## 依設計的安全性

透過整個開發程序將安全性納入考量的系統工程方法。

### 安全控制

一種技術或管理防護機制，它可預防、偵測或降低威脅行為者利用安全漏洞的能力。安全控制有四種主要類型：[預防性](#)、[偵測性](#)、[回應性](#)和[主動性](#)。

### 安全強化

減少受攻擊面以使其更能抵抗攻擊的過程。這可能包括一些動作，例如移除不再需要的資源、實作授予最低權限的安全最佳實務、或停用組態檔案中不必要的功能。

### 安全資訊與事件管理 (SIEM) 系統

結合安全資訊管理 (SIM) 和安全事件管理 (SEM) 系統的工具與服務。SIEM 系統會收集、監控和分析來自伺服器、網路、裝置和其他來源的資料，以偵測威脅和安全漏洞，並產生提醒。

### 安全回應自動化

預先定義和程式設計的動作，旨在自動回應或修復安全事件。這些自動化可做為[偵測](#)或[回應](#)式安全控制，協助您實作 AWS 安全最佳實務。自動化回應動作的範例包括修改 VPC 安全群組、修補 Amazon EC2 執行個體或輪換登入資料。

### 伺服器端加密

由 AWS 服務接收資料的 在其目的地加密資料。

### 服務控制政策 (SCP)

為 AWS Organizations 中的組織的所有帳戶提供集中控制許可的政策。SCP 會定義防護機制或設定管理員可委派給使用者或角色的動作限制。您可以使用 SCP 作為允許清單或拒絕清單，以指定允許或禁止哪些服務或動作。如需詳細資訊，請參閱 AWS Organizations 文件中的[服務控制政策](#)。

### 服務端點

的進入點 URL AWS 服務。您可以使用端點，透過程式設計方式連接至目標服務。如需詳細資訊，請參閱 AWS 一般參考中的 [AWS 服務端點](#)。

### 服務水準協議 (SLA)

一份協議，闡明 IT 團隊承諾向客戶提供的服務，例如服務正常執行時間和效能。

### 服務層級指標 (SLI)

服務效能方面的測量，例如其錯誤率、可用性或輸送量。

## 服務層級目標 (SLO)

代表服務運作狀態的目標指標，由[服務層級指標](#)測量。

## 共同責任模式

描述您與共同 AWS 承擔雲端安全與合規責任的模型。AWS 負責雲端的安全，而負責雲端的安全。如需詳細資訊，請參閱[共同責任模式](#)。

## SIEM

請參閱[安全資訊和事件管理系統](#)。

## 單點故障 (SPOF)

應用程式的單一關鍵元件故障，可能會中斷系統。

## SLA

請參閱[服務層級協議](#)。

## SLI

請參閱[服務層級指標](#)。

## SLO

請參閱[服務層級目標](#)。

## 先拆分後播種模型

擴展和加速現代化專案的模式。定義新功能和產品版本時，核心團隊會進行拆分以建立新的產品團隊。這有助於擴展組織的能力和服務，提高開發人員生產力，並支援快速創新。如需詳細資訊，請參閱[中的階段式應用程式現代化方法 AWS 雲端](#)。

## SPOF

請參閱[單一故障點](#)。

## 星狀結構描述

使用一個大型事實資料表來存放交易或測量資料的資料庫組織結構，並使用一或多個較小的維度資料表來存放資料屬性。此結構旨在用於[資料倉儲](#)或商業智慧用途。

## Strangler Fig 模式

一種現代化單一系統的方法，它會逐步重寫和取代系統功能，直到舊式系統停止使用為止。此模式源自無花果藤，它長成一棵馴化樹並最終戰勝且取代了其宿主。該模式由[Martin Fowler 引入](#)，作

為重寫單一系統時管理風險的方式。如需有關如何套用此模式的範例，請參閱[使用容器和 Amazon API Gateway 逐步現代化舊版 Microsoft ASP.NET \(ASMX\) Web 服務](#)。

## 子網

您 VPC 中的 IP 地址範圍。子網必須位於單一可用區域。

## 監控控制和資料擷取 (SCADA)

在製造中，使用硬體和軟體來監控實體資產和生產操作的系統。

## 對稱加密

使用相同金鑰來加密及解密資料的加密演算法。

## 合成測試

以模擬使用者互動的方式測試系統，以偵測潛在問題或監控效能。您可以使用 [Amazon CloudWatch Synthetics](#) 來建立這些測試。

## 系統提示

一種向 [LLM](#) 提供內容、指示或指導方針以指示其行為的技術。系統提示有助於設定內容，並建立與使用者互動的規則。

# T

## 標籤

做為中繼資料以組織 AWS 資源的鍵值對。標籤可協助您管理、識別、組織、搜尋及篩選資源。如需詳細資訊，請參閱[標記您的 AWS 資源](#)。

## 目標變數

您嘗試在受監督的 ML 中預測的值。這也被稱為結果變數。例如，在製造設定中，目標變數可能是產品瑕疵。

## 任務清單

用於透過執行手冊追蹤進度的工具。任務清單包含執行手冊的概觀以及要完成的一般任務清單。對於每個一般任務，它包括所需的預估時間量、擁有者和進度。

## 測試環境

請參閱 [環境](#)。

## 訓練

為 ML 模型提供資料以供學習。訓練資料必須包含正確答案。學習演算法會在訓練資料中尋找將輸入資料屬性映射至目標的模式 (您想要預測的答案)。它會輸出擷取這些模式的 ML 模型。可以使用 ML 模型，來預測您不知道的目標新資料。

## 傳輸閘道

可以用於互連 VPC 和內部部署網路的網路傳輸中樞。如需詳細資訊，請參閱 AWS Transit Gateway 文件中的[什麼是傳輸閘道](#)。

## 主幹型工作流程

這是一種方法，開發人員可在功能分支中本地建置和測試功能，然後將這些變更合併到主要分支中。然後，主要分支會依序建置到開發環境、生產前環境和生產環境中。

## 受信任的存取權

將許可授予您指定的服務，以代表您在組織中 AWS Organizations 及其帳戶中執行任務。受信任的服務會在需要該角色時，在每個帳戶中建立服務連結角色，以便為您執行管理工作。如需詳細資訊，請參閱文件中的 AWS Organizations [搭配使用 AWS Organizations 與其他 AWS 服務](#)。

## 調校

變更訓練程序的各個層面，以提高 ML 模型的準確性。例如，可以透過產生標籤集、新增標籤、然後在不同的設定下多次重複這些步驟來訓練 ML 模型，以優化模型。

## 雙比薩團隊

兩個比薩就能吃飽的小型 DevOps 團隊。雙披薩團隊規模可確保軟體開發中的最佳協作。

# U

## 不確定性

這是一個概念，指的是不精確、不完整或未知的資訊，其可能會破壞預測性 ML 模型的可靠性。有兩種類型的不確定性：認知不確定性是由有限的、不完整的資料引起的，而隨機不確定性是由資料中固有的噪聲和隨機性引起的。如需詳細資訊，請參閱[量化深度學習系統的不確定性](#)指南。

## 未區分的任務

也稱為繁重工作，是建立和操作應用程式的必要工作，但不為最終使用者提供直接價值或提供競爭優勢。未區分任務的範例包括採購、維護和容量規劃。

## 較高的環境

請參閱 [環境](#)。

## V

### 清空

一種資料庫維護操作，涉及增量更新後的清理工作，以回收儲存並提升效能。

### 版本控制

追蹤變更的程序和工具，例如儲存庫中原始程式碼的變更。

### VPC 對等互連

兩個 VPC 之間的連線，可讓您使用私有 IP 地址路由流量。如需詳細資訊，請參閱 Amazon VPC 文件中的 [什麼是 VPC 對等互連](#)。

### 漏洞

危及系統安全性的軟體或硬體瑕疵。

## W

### 暖快取

包含經常存取的目前相關資料的緩衝快取。資料庫執行個體可以從緩衝快取讀取，這比從主記憶體或磁碟讀取更快。

### 暖資料

不常存取的資料。查詢這類資料時，通常可接受中等緩慢的查詢。

### 視窗函數

SQL 函數，對與目前記錄在某種程度上相關的資料列群組執行計算。視窗函數適用於處理任務，例如根據目前資料列的相對位置計算移動平均值或存取資料列的值。

### 工作負載

提供商業價值的資源和程式碼集合，例如面向客戶的應用程式或後端流程。

## 工作串流

遷移專案中負責一組特定任務的功能群組。每個工作串流都是獨立的，但支援專案中的其他工作串流。例如，組合工作串流負責排定應用程式、波次規劃和收集遷移中繼資料的優先順序。組合工作串流將這些資產交付至遷移工作串流，然後再遷移伺服器 and 應用程式。

## WORM

請參閱[寫入一次，讀取許多](#)。

## WQF

請參閱[AWS 工作負載資格架構](#)。

## 寫入一次，讀取許多 (WORM)

儲存模型，可一次性寫入資料，並防止資料遭到刪除或修改。授權使用者可以視需要多次讀取資料，但無法變更資料。此資料儲存基礎設施被視為[不可變](#)。

## Z

### 零時差入侵

利用[零時差漏洞](#)的攻擊，通常是惡意軟體。

### 零時差漏洞

生產系統中未緩解的瑕疵或漏洞。威脅行為者可以使用這種類型的漏洞來攻擊系統。開發人員經常因為攻擊而意識到漏洞。

### 零鏡頭提示

提供 [LLM](#) 執行任務的指示，但沒有可協助引導任務的範例 (快照)。LLM 必須使用其預先訓練的知識來處理任務。零鏡頭提示的有效性取決於任務的複雜性和提示的品質。另請參閱[少量擷取提示](#)。

### 殭屍應用程式

CPU 和記憶體平均使用率低於 5% 的應用程式。在遷移專案中，通常會淘汰這些應用程式。

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。